IN THE UNITED STATES DISTRICT COURT

FOR THE DISTRICT OF MARYLAND

NOVELL INC., Plaintiff, v. MICROSOFT CORPORATION, Defendant.	:	MDL Docket No. 1332 **Confidential Expert Report of John K. Bennett, Ph.D., P.E. June 19, 2009**

Table of Contents

1. ENGAGEMENT 3

2. QUALIFICATIONS 4

2.A. Professional Background 4

2.B. Publications Authored in the Preceding Ten Years 9

2.C. Cases in Which I Have Testified at Trial or by Deposition as an Expert 9

3. MATERIALS REVIEWED 10

4. SUMMARY OF EXPECTED TESTIMONY 10

5. DISCUSSION OF ISSUES 12

5.A. Relevant Issues in Software Design 12

5.A.1. Constraints on Software Design 15

5.A.2. The Evolution of Operating System Design 18

5.A.3. Data Abstraction and Modularity 20

5.A.4. System Software: Definition of the System Software Layer 24

5.A.5. Middleware 27

5.A.6. Operating System Graphical User Interface 28

5.A.7. Integration of Functionality Into the Operating System 31

6. ALLEGATIONS RAISED IN THE REPORT OF PLAINTIFF'S EXPERT 31

6.A. General Observations 31

6.A.1. Lack of Basis, Foundation, Meaningful Citation in Alepin Report 31

6.A.2. Conflation of Unrelated Issues in Alepin Report 39

6.A.3. Mr. Alepin's Reliance on Undocumented Interviews 41

6.A.4. Self Citation in Alepin Report 41

6.A.5. The Need for Precision When Defining Terms 41

6.B. Allegations Regarding "Shell Namespace Extensions" 43

6.B.1. Context 46

6.B.2. Change is a Normal and Expected Part of the Software Development Process 50

6.B.3. Technical Basis for Not Formally Documenting Shell Namespace Extensions 51

6.B.4. Efforts to Mitigate the Effect of Decision on ISVs 53
6.B.5. WordPerfect Coding for Windows 95 had not Started 54
6.B.6. No High-Level Complaints to Microsoft 58
6.B.7. Office Applications Did Not Use the Functionality in Question 60
6.B.8. Options Available to Developers 60
6.B.9. WordPerfect Had Problems Unrelated to Alleged Microsoft Conduct 66
6.B.10. Harral and Richardson Deposition Testimony 68
6.C. Allegations Regarding MAPI 69
6.C.1. What is MAPI? 69
6.C.2. MAPI-Related Allegations 73
6.C.3. Microsoft Outlook/Exchange Protocols 74
6.C.4. Allegations Regarding Groupware 75
6.D. Allegations Regarding Windows 95 Logo Certification 81
6.D.1. Novell's Request for Exemption 90
6.E. Allegations Regarding Printing Interfaces in Windows 95 94
6.F. Miscellaneous Allegations in Alepin Report 99

1. Engagement

1. This report contains a summary of my opinions, together with my research and analysis to date, in connection with *Novell Inc. v. Microsoft*. I have been retained by counsel for Defendant Microsoft as an expert witness in the above entitled action. I expect to be called as an expert witness if and when this case comes to trial. As I continue my work on the issues raised in this case, I may supplement, refine or revise my opinions and findings as a result of further review and analysis.

2. In November 1997, I was retained by counsel to Microsoft to evaluate certain allegations raised by Caldera, Inc. in connection with the case *Caldera, Inc.* v. *Microsoft Corporation* (the "Caldera" case), and in that connection, I reviewed the extensive factual record developed during discovery in that case and reached certain opinions. In May 2000, and again in December 2001, I was asked by counsel to Microsoft to evaluate certain claims relating to allegations made by the Antitrust Division of the U.S. Department of Justice and certain State Attorneys General (the "Washington DC" case). In June 2001, I was asked by counsel to Microsoft to evaluate certain allegations made initially in the Caldera and Washington DC cases, and which were alleged again in consolidated civil actions in California and Maryland (the "California" and "MDL" cases, respectively), as well as other allegations asserted by the class action plaintiffs in those actions. In June 2003, I was asked to evaluate certain of the allegations made initially in the Caldera, Washington DC, MDL and California cases, and which were alleged again in civil actions in Arizona and Minnesota (the "Arizona" and "Minnesota" cases, respectively). In September, 2004, I was asked by counsel to Microsoft to evaluate certain claims relating to

technology associated with Internet streaming and Java, and other technical issues, in connection with a case captioned *Burst.com, Inc. v. Microsoft Corporation* (the "Burst" case).

3. In March 2006, I was asked to evaluate, again, certain of the allegations made initially in the Caldera, Washington DC, MDL, California, Arizona, Minnesota and Burst cases, in connection with a case captioned *Comes et al.* v. *Microsoft* (the "Iowa" case). In August 2008, I was asked to evaluate, again, certain of the allegations made initially in the Caldera, Washington DC, MDL, California, Arizona, Minnesota, Burst and Iowa cases, in connection with a case captioned *Jim Hood, Attorney General ex rel. State of Mississippi* v. *Microsoft* (the "Mississippi" case). In March 2009, I was asked to evaluate allegations made in *Novell Inc.* v. *Microsoft* (the "Novell" case).

4. I am being compensated at a rate of $600 per hour for my time in connection with my analysis. My compensation is not in any way dependent upon the outcome of this litigation.

2. Qualifications

2.A. Professional Background

5. I am the Archuleta Professor of Computer Science and the Director of the ATLAS Institute at the University of Colorado at Boulder. I hold joint faculty appointments in Computer Science and in Electrical and Computer Engineering.

6. I received a Bachelor of Science in Electrical Engineering (1973) and a Master of Electrical Engineering (1974) from Rice University in Houston, Texas. Some years later, I received a

Master of Science degree (1983) and a Ph.D. (1988) in Computer Science from the University of Washington in Seattle, Washington.

7. For the past thirty-plus years, I have focused on the development of computer software, both directly as a creator of sophisticated software applications and operating systems, and as an academic, studying and contributing to the field.

8. I began programming computers in 1969 as an undergraduate at Rice University. My formal training at Rice concentrated in the area of computer system hardware and software design. While I was an undergraduate at Rice University, I obtained part-time (20 hours/week during the school year, full time during the summer) employment with Texas Scientific Corporation, first as an engineering assistant, and later as a hardware designer and programmer. During my senior year, I assumed responsibility for all prototype testing and evaluation of the company's products. Examples of projects on which I worked included the mission control system for Jet Propulsion Laboratories and the control system for the Washington, D.C. Metropolitan Transit Authority light rail system.

9. While I was obtaining my Masters of Electrical Engineering degree at Rice, I was also employed as a research and teaching assistant. During that time, I participated in the design, construction, programming and checkout of the R2 Rice Research Computer, a well-known project that pioneered what are called "tagged architectures".

10. After obtaining my Masters of Electrical Engineering degree from Rice, I was commissioned as an officer in the U.S. Navy. My initial assignment in the Navy was as the

Electrical Officer of a new gas-turbine-powered destroyer. In this capacity, I managed the testing and support of the ship's computerized propulsion control system. This assignment involved the identification, documentation and correction of hundreds of design deficiencies in this new system. I later became a designated Engineering Duty Officer, and was assigned to Puget Sound Naval Shipyard. At PSNS, I served as the Senior Ship Superintendent responsible for all aspects of the repair and overhaul of two nuclear cruisers and a conventional destroyer.

11. After completing my service in the Navy, I entered graduate school in the Department of Computer Science at the University of Washington, where I received my second Masters degree and my Doctoral degree, both in Computer Science. While at the University of Washington, I took advanced coursework in computer system architecture, software engineering, and operating system design. I was instrumental in the design and development of the Eden Distributed Computer System, one of the first object-oriented distributed computing systems ever developed. My doctoral dissertation involved the design, implementation and evaluation of another object-oriented distributed computing system based upon Smalltalk. A predecessor of Sun's Java technology, Smalltalk is an object-oriented programming environment originally developed at the Xerox Palo Alto Research Center.

12. Prior to completing my Doctorate at the University of Washington, I founded and served as President of Pacific Mountain Research, Inc. ("PMR") in Seattle, Washington. At PMR, I directed the design and development of hardware and software components of a variety of commercial computer systems including: (1) multi-processor and virtual memory systems, (2) several high performance graphics and image processing systems, (3) custom very large scale

integrated (VLSI) circuits, (4) biomedical instrumentation, and (5) video processing systems. PMR designs were utilized or marketed by several companies, including Motorola, Signetics, RCA, IBM, Eastman Kodak, Microsoft, Sequent, Weyerhaeuser, the Department of Defense, and the New York Stock Exchange. PMR was frequently responsible for the testing and initial software and hardware support of these systems.

13. After completing my Doctorate, I joined the faculty of the Department of Electrical and Computer Engineering at Rice University in 1988. At Rice, my research focused broadly in the area of operating system/microprocessor architecture interaction and, more narrowly, in the area of distributed and parallel computing systems. With my students, I developed the Munin Distributed Shared Memory ("DSM") system, the prototype for most modern software DSM systems, and the Brazos Parallel Programming Environment, which supports clusters of multiprocessor computers (multicomputers) running Windows NT and Windows 2000. Brazos was used by a number of academic and research users around the world. My students and I also developed several mechanisms that exploit high-performance user-level networks. One aspect of this work was the implementation of a better-performing alternative to Microsoft's Windows Sockets Direct Path, a high-performance networking protocol. Other work I did at Rice concentrated on building efficient and reliable execution platforms for parallel applications on clusters of multicomputers, fault-tolerant parallel robotics, and detecting holes in surgical gloves during use. During my last four years at Rice, I served as the Master of Richardson College.

14. In the summer of 2000, I joined the faculty in the Departments of Computer Science and Electrical and Computer Engineering at the University of Colorado at Boulder, where I have

taught, among other subjects, the graduate Advanced Operating Systems class and the graduate Hybrid Embedded Systems class. In 2002, I was appointed Associate Dean of Engineering at the University of Colorado at Boulder; in 2007, I was named Director of the ATLAS Institute, a campus-wide effort to support innovative interdisciplinary initiatives involving information and communication technology.

15. During my academic career, I have published over 60 papers and I have supervised 16 doctoral dissertations and masters theses, including work related to operating system design. I have received significant research funding from industry and government organizations in support of my research in over 30 research and educational projects. I have been an active member of the Association for Computing Machinery since 1979, and the Institute for Electrical and Electronic Engineers (in which I hold the rank of Senior Member) since 1976. I am a registered Professional Engineer in the State of Texas. In fall 2003, I was asked by the President of the National Academy of Science to join him in representing the United States on a goodwill mission to the Republic of Korea to exchange ideas about science and engineering education. I was elected to membership in the European Academy of Sciences in 2002.

16. As a result of this experience, I am thoroughly familiar with hardware and software development, as well as software testing and support. I have participated in the development and testing of commercial software products as both a software developer and manager. In the course of my research and teaching, I have also become broadly familiar with various client and server operating systems, and a variety of applications development and enterprise programming environments.

17. Over the past 18 years, I have been qualified as an expert in computer systems in courts throughout the United States, and have given testimony on subjects relating to computer software in more than a dozen cases. My expert testimony has included analysis of computer software behavior, comparisons of functionally similar software and hardware, patent claim interpretation and patent claim analysis. I am a named inventor on two U.S. Patents.

18. A summary of my qualifications is described in the attached resume (Appendix A).

2.B. Publications Authored in the Preceding Ten Years

19. The publications I have written in the last ten years are set forth in Appendix A.

2.C. Cases in Which I Have Testified at Trial or by Deposition as an Expert

20. I have testified as an expert in the following cases:

Haden and Company v. AMS (1990)
ResTech v. Joseph Hawkins (1993)
Siege Industries, Inc. v. Clark Manufacturing (1996)
Caldera, Inc. v. Microsoft Corporation (1997)
Bristol Technology, Inc. v. Microsoft Corporation (1999)
Compaq Computer Corporation v. eMachines Inc. (2000)
Compaq Computer Corporation v. Unova, Inc. et al. (2001)
State of New York ex. rel. v. Microsoft Corporation (2002)
Microsoft California consumer cases (2002)
Microsoft MDL consumer cases (2002)
Friedman v. Microsoft (2003)
Gordon v. Microsoft (2003)
New River Holding Limited Partnership v. Precision Response Corporation, et al. (2004)

SuperSpeed Software, Inc. v. Oracle Corporation (2005)
Comes et al. v. Microsoft Corporation (2006)
Paltalk Holdings, Inc v. Microsoft Corporation (2008)
SuperSpeed Software, Inc. v. IBM Corporation (2008)

21. I have also provided consulting and expert opinions not involving testimony to several other companies. This other work is in general governed by protective orders or non-disclosure agreements that prevent me from disclosing information related to those engagements.

3. Materials Reviewed

22. My knowledge in these areas (and, therefore, my opinions) derives in part from my active involvement in computer systems research and development for the past thirty plus years. For twenty of those years, I have taught classes addressing computer system design and development issues (including those relating to operating systems), and as part of my teaching and academic advancement, I keep current on issues in computer software, including (specifically) operating system development.

23. Aside from my background in computer science, in the course of preparing this report I have taken into consideration the materials enumerated in Appendix B. In addition, I have considered publicly available documents relating to issues in the case, including press articles, books, and web sites, and other documents and information specifically cited in this report.

4. Summary of Expected Testimony

24. I expect to testify on the following subjects:

a) To respond to those technical matters discussed in the report of Plaintiff's expert Ronald Alepin (the "Alepin Report"), and in particular those matters where I believe his conclusions to be wrong, or his discussion to be inaccurate or misleading; and

b) To respond to technical matters raised in the testimony of percipient witnesses, to the extent I have opinions that are responsive.

25. I expect to supplement my views, should additional information or materials relevant to this case be made available to me, including information disclosed during the depositions of the Plaintiff's experts or otherwise.

26. In the preparation of my report, I have attempted to consider the entire relevant evidentiary record in forming my opinions related to the issues raised in this case. However, as a general matter, there are certain issues on which I cannot offer opinions. For example, it appears to me that Plaintiff's expert Mr. Alepin has confused the function and behavior of a software program (the *technical purpose*) with the corporate strategic goals and objectives, as well as the broader host of non-technical factors that contribute to the business environment in which software is developed and sold, and that may have led to the development of that software (the *business purpose*). The discipline of computer science does not address the latter topic.

27. In his Expert Report of May 1, 2009, Mr. Alepin offered a number of opinions of a non-technical nature, including opinions regarding Microsoft's alleged business practices and corporate intent. Since these issues fall outside of my technical expertise, I offer no opinion on those issues in my report.

28. In his report, Mr. Alepin also offered certain opinions for which no basis or foundation was apparent, as I describe below. I am unable to respond to these opinions without an identification of the technical foundation underlying Mr. Alepin's stated views on these issues, but may have conclusions and testimony bearing on these issues to the extent that Mr. Alepin later provides some basis or explanation, in his deposition, or otherwise.

5. Discussion of Issues

5.A. Relevant Issues in Software Design

29. Writing software is a complex and labor-intensive enterprise, as discussed in further detail below. When creating new software (or when modifying existing software), the software developer must take into account various constraints:

- The target hardware architecture, including both the computer itself, as well as any associated peripheral devices;
- The available software development platforms and the functionality exposed by those platforms to other software;
- The intended use and intended base of users of the software under development;
- Whether the software is intended to provide functionality to other software by exposing an application program interface ("API"), and whether the other software is "trusted" (known to behave in a reliable manner) or "untrusted" (in which case every attempt by other software to access the functionality provided by the API must be verified for correct behavior);

- Whether the software will use functionality provided by APIs other than those exposed by the relevant software development platform;
- The mechanisms by which the software under development will communicate with other software, both that installed on the same computer and that installed on other computers;
- Whether the software must be backward-compatible with other software that may already be in use on the computer;
- Whether the software will take advantage of sophisticated, high-level functionality provided by the underlying platform, or whether the software will only access basic, low-level functionality;
- Whether the software must be protected from unauthorized or unsafe access, and whether the software will employ security mechanisms in order to gain access to other software; and
- Non-technical constraints including time to market and budget limitations such as those imposed by customer demand or the existence of competing software.

30. Virtually all of these constraints come into play when the software in question is an operating system or a component of an operating system. Further, because an operating system by design provides services to other software, the details of which are generally not known, and are inherently unknowable, operating system designers take steps to ensure that application software only interacts with the operating system in well-defined ways. For this purpose, operating system developers typically publish APIs that expose a set of system services intended

for use by application programs.[1] When an application program makes a "call" (an invocation of some specific functionality) to one of these system services (an API), the operating system performs a number of checks to ensure that the call is valid. These checks typically include:

- whether the caller is authorized to make the indicated call;
- whether all of the parameters passed with the call are valid (e.g., does a parameter that is supposed to point to memory in fact point to a valid region of memory?);
- whether the parameters are all of the correct type (e.g., is a parameter that is supposed to be an integer actually an integer?); and
- whether it is safe to make the changes in system state that executing the call will involve (e.g., if the call involves writing to a hardware device, has that hardware device completed processing the last call?).

31. These checks are necessary to ensure that the operating system will continue to function correctly, but necessarily take time to perform. If the behavior of calling software is known completely - for example, if the calling software is another component of the operating system - then it may be possible to "short-circuit" some of these checks in order to improve the

[1] In his report, Mr. Alepin routinely uses the term "API" to refer to any programming interface. While it is true that the term "API" is frequently misused in casual conversation to refer to any programming interface, such misuse does not change the true definition of the term. The words "Application Programming Interface", as the name confirms, correctly refers to those programming interfaces that are exposed to and used by application programs. Programming interfaces used solely within the operating system would correctly be referred to as "System Programming Interfaces". Mr. Alepin's use of the term "undocumented API", and his discussion of "undocumented interfaces" in general, (Alepin Report at 32) is particularly problematic, and appears to reflect a flawed understanding of this issue. Every modern operating system has literally thousands of internal interfaces that are not exposed to application programs for good engineering reasons, as I explain later in this report.

performance of the operating system. I stress that this is only possible if it is absolutely certain that the calling software will behave in a known way, thereby rendering the checks unnecessary.

5.A.1. Constraints on Software Design

32. Microprocessor architectures impose physical limitations on the software they support. Intel, Motorola, AMD and other semiconductor manufacturers have created elaborate designs that impose practical limits on what it is possible to do on a particular microprocessor (or "chip"). For example, the Intel 8088/8086 microprocessor could not address more than one megabyte of random access memory ("RAM"). Software either had to conform to this physical limitation of the microprocessor, or rely upon other hardware and associated software to circumvent this limitation.

33. In connection with a discussion of these physical limitations, it is helpful to review certain terminology and technology related to Intel microprocessors and Microsoft Windows.

34. The Intel microprocessors on which most Microsoft operating systems run are all members of the "x86" family of processors. This family includes the 8088 (which was the microprocessor used in the first IBM PC), the 8086, 80286, 80386, 80486, and the Pentium and Itanium series of processors. Early processors like the 8088 and 8086 had very limited support for operating system software. In particular, they were capable of addressing only one megabyte of memory (on early IBM PC's the upper 384 kilobytes of this one megabyte were used to support peripheral devices, so in actuality, a maximum of only 640 kilobytes of memory were available). These early processors did not provide support for virtual memory (the ability to access a

"virtual" address space larger than the actual memory present) or multitasking (the ability of the microprocessor to execute two or more tasks simultaneously). Later processors are capable of operating in a mode compatible with the limitations of their early precursors. This 8086-compatible mode is called "real mode". Modern Intel processors operate in "protected mode" in order to access memory beyond the one-megabyte boundary and to support multitasking and virtual memory. One of the features of protected mode operation is the ability to create several "virtual" 8086 processors that are able to operate independently of one another on the same microprocessor. This mode is called "virtual 8086 mode" or "virtual real mode".

35. Windows and MS-DOS also run in "modes" that indicate the latest microprocessor features supported by that mode. Thus, MS-DOS runs in "Real mode", indicating that the processor running MS-DOS is in real mode (note the use of case to distinguish between processor modes and Operating System Modes - I will use this convention throughout this report). Windows has two other operating system modes: Standard and Enhanced (sometimes referred to as "386 Enhanced", since an 80386 or later processor was required to support this mode).

36. In "Standard mode", the operating system is able to address memory beyond the one-megabyte boundary and utilize the protection features available on the 80286 processor. Thus, Windows in Standard mode is running on a processor in protected mode. Only an 80286 or later processor can run Windows in Standard mode.

37. In "Enhanced mode" (which runs on an 80386 or later processor in protected mode), the operating system is able to take advantage of the full virtual memory and multitasking capa-

bilities of the processor, as well as to support multiple virtual machines for legacy DOS programs.

38. Microprocessor designers may also incorporate unique features, while retaining backward compatibility. For example, Intel developed a variety of hardware features and low-level machine instructions that supported multi-media applications. These instructions, called "MMX Instructions", facilitate the processing of an array of data using a single instruction. This kind of instruction is often referred to as "Single Instruction Multiple Data", or simply "SIMD". SIMD instructions are particularly useful for computations needed in processing graphics, video and audio data. MMX Instructions have been emulated by other microprocessor manufacturers, including AMD and Cyrix.

39. Apart from these physical limitations on software design, there are environmental limitations imposed by the possible presence of other software expected to run simultaneously with the software under development. Unless the software developer is aware of the presence of this other software, he or she may not know of these limitations. For example, particularly for programs intended to operate in real mode, I could write a software program (called "MyProgram") that assumed it could use some portion of memory within the computer. If, however, that section of memory is already being used by another program (a fact I might not know) ("TheirProgram"), it is possible that MyProgram might fail to work as I expect. It is also possible that TheirProgram might be interfered with, even though I did not know in advance that MyProgram would have any impact on TheirProgram.

40. The situation is even more complicated if there is a conflict between an existing program and a new program I am writing: I'll use the example of OldWare and MyNewWare. A technique I might wish to employ in MyNewWare may be inconsistent with the way in which OldWare works. I would then have (at least) two choices: I could write software code in MyNewWare to work around (and not touch) the code I find to be "inconsistent" with the code contained in OldWare. Another approach might be for MyNewWare to "patch" or change the pre-existing and "incompatible" software code of OldWare (after it is loaded into memory) to something that is "compatible" with the new technique I am proposing to use. This technique, called software "patching", is possible because once a program is loaded into memory it can be modified in much the same way as one might modify a word processing document. If MyNewWare patches OldWare in this manner, others who previously wrote software code just for "OldWare" might have to change their products to work with "MyNewWare." These changes may or may not be easy to implement. However, these types of adjustments to accommodate other software products have been quite common in the computer software industry.

5.A.2. The Evolution of Operating System Design

41. Operating systems began as simple programs that did little more than load an application program into memory. This was a simple operation, because early computers had only a single processor, performed only one task at a time, had very few peripheral devices, and had very simple communication requirements. Since memory was very limited, early computer users did not want to devote a significant amount of the very limited memory available to the operating

system. Operating systems for microprocessor-based computers had similarly humble beginnings, but have evolved over the last thirty years or so from very simple programs with very limited functionality to the fully-functioned operating systems to which users of personal computers today are accustomed. In part, this evolution has been fueled by the increasing capabilities and performance of hardware, but in large part the evolution of operating systems has tracked the need for the operating system to offer ever increasing functionality to application programs.

42. The presence of rich functionality in operating systems facilitates the creation of rich applications and helps provide a common user experience across applications. Indeed, application software developers and the users of their applications have to be able to rely upon the presence of needed operating system functionality. This rich functionality, present in the operating system, in turn makes it possible for developers and thus a broader class of users to make productive use of computers.

43. In the course of the evolution of operating systems, their design requirements have over time undergone significant alteration. These changes in design requirements have required corresponding fundamental changes in the way that operating systems were developed. For example, when operating systems became too complex to manage as a single monolithic program, operating systems had to be designed in a modular fashion. When it became important to protect the execution of the operating system from ill-behaved application programs, the principles of modularity and data abstraction discussed below became critical to the performance and reliability of the operating system.

44. The limited amount of memory in personal computers has been a major design constraint for most operating systems. In addition, as operating systems have become more complex, the effort required to develop, debug, test, maintain and modify operating system source code has increased significantly. These two constraints led operating system developers (actually, all software developers) to employ the principles of *code sharing* and *code reuse* in their designs. Code sharing refers to the practice of several software modules using a single block of code to implement required functionality that they have in common, in contrast to each module implementing the required functionality individually. Code reuse typically refers to the practice of using the existing code base to the extent possible when developing new software. In general, code reuse is much easier if the software in question has been designed with code reuse in mind.

45. Employing code sharing and code reuse is a sound software engineering practice, for three reasons: (1) it lessens the presence of redundant code, thereby shrinking the "memory footprint" of the software being implemented; (2) it makes the software easier to maintain, because when it is time to update or modify the software, the functionality of interest will be found in just one location in the code, rather than being replicated in many places in the code; and (3) it offers consistency of operation because identical functions invoked in different places in the software are performed by the same code.

5.A.3. Data Abstraction and Modularity

46. Some application developers, and others, have argued that Microsoft should routinely publish information describing the internal workings of its operating systems, including large numbers of internal interfaces that were never meant for external use. There are sound technical

reasons for operating system vendors *not* to make such information public, which I summarize below.

47. During the 1980's, the principles of data abstraction and modular programming came to be recognized within the computer science community as offering programmers the ability to develop complex software that was easier to debug and maintain. Data abstraction refers to the technique of hiding internal data representations, and exposing only a well-defined external interface for use by others. Modular programming refers to the "divide and conquer" software engineering practice of dividing large software projects into a collection of "modules" that have well-defined "interfaces". These modules are much easier to develop separately than a larger monolithic program. Further, as long as the interfaces do not change, the internal details of the individual modules only need to be understood within that module, again making the software easier to develop and modify.

48. For operating systems, it is also common to expose two versions of module interfaces, a private one for other software components that are "trusted", *i.e.*, known to behave in a particular manner, and a public one for "untrusted" components that cannot be presumed to be so well behaved. Trusted components would typically be other parts of the operating system, or application software for which the behavior is well known. Untrusted components would typically be third-party programs written to call upon operating system services, such as application and utility programs. It is also possible for operating systems to employ a spectrum of trust relationships between the components of the operating system, forming a hierarchy of privilege. Every modern operating system with which I am familiar, including many versions of

Windows and Sun's version of Unix called Solaris, employs some variant of this distinction between trusted and untrusted software components.

49. Applying the approach just described to operating system development, calls that directly access critical operating system components would typically be part of the private interface; calls that indirectly access critical operating system components in a restricted manner would typically be part of the public interface. One of the advantages of this approach to software development is that the internal details of data representation are hidden from the user of that data representation. The operating system developer is then free to make changes (*i.e.*, improvements) to this internal data representation, as long as the external interfaces relied upon by third parties are preserved. Further, if the number of modules using the private interface is small, it is easier to make changes in the private interface in order to improve system performance or reliability, or to add new functionality.

50. There is also an advantage to having information about the private interface be closely held. Once published, it is difficult to change the public interface used to access operating system services. This is because programmers who have written code that relies on that public interface expect it not to change, because such change would adversely impact their application. Significant changes to the public interface may even necessitate a complete rewriting of the original application program, a costly and time-consuming undertaking. In the case of recent operating systems developed by Microsoft, there are, conservatively, thousands of such application programs relied upon by millions of end users.

51. A related concern occurs when some private function or data structure is discovered, made public, and used by significant numbers of programmers. There are many individuals who have made it a practice to discover and publish these sorts of details, and the many books on the subject attest to the willingness of some software developers to take advantage of such information. Widespread use of some part of the private interface (without the operating system developer's intent) effectively moves this portion of the private interface into the public interface. The negative consequence of this practice is that the operating system developer now may be faced with a choice between improving performance, which requires a change to the private interface (and which the original designer may have had every intention of changing in the future), or maintaining compatibility with a de facto standard never intended as a permanent part of the operating system specification.

52. Early on, Microsoft published clear warnings to application programmers that the use of undocumented functions is a dangerous practice. Microsoft's KnowledgeBase article Q34761 of September 5, 1988 entitled "Regarding the Use of Undocumented MS-DOS Functions" states:

> "Microsoft does not give out any information about undocumented system features. If calls, flags, or interrupts are undocumented, it is because they are not supported; we can give NO guarantee that they will exist in future releases of DOS. If you find out about these features (through articles or by chance) and begin using then in your programs, there is a real potential that your application will not work in future DOS versions. We strongly advise against using undocumented features for these reasons and will give out no information about their use."

53. Well-disciplined application programmers do not use undocumented functions of an operating system. Following this policy is a good practice that benefits all concerned: (1) the operating system developer, who is thereby free to improve the performance of its operating

systems (without stranding users and developers by not accommodating undocumented APIs), (2) the application developer, who can now rely on his or her program working with future versions of the operating system, and (3) the end user, who can upgrade his or her operating system without fear of breaking important applications. This policy even benefits a potential operating system cloner, who now has a clear specification of the functionality that he or she needs to emulate. Unfortunately, this policy has been frequently ignored by application programmers writing for Microsoft operating systems, with the consequence of negating the advantages noted above. Adherence to this policy by software developers within and outside of Microsoft is discussed below.

5.A.4. System Software: Definition of the System Software Layer

54. Mr. Alepin's depiction of the "system software layer" (Alepin Report at 21), and his subsequent description of the components of this "layer" is in my view simplistic, and does not accurately reflect the structure or organization of modern operating systems in general, or Windows in particular. The components of an operating system are *not* simple isolated "boxes" of functionality as Mr. Alepin's figure suggests. Instead, operating system components are typically highly interdependent software modules that together provide increasingly general layers of resource abstraction to higher-level software. These interdependencies, together with the typical need to maintain backward compatibility with prior versions, constrain operating system design in ways apparently not envisioned by Mr. Alepin. Figure 1 below helps illustrate these ideas. In Figure 1, I have depicted just a few of the key software modules in modern Windows operating systems that expose APIs to application programs and other modules of the

operating system. Arrows on the diagram depict call dependencies, *i.e.*, if an arrow is drawn from user32.dll to kernel32.dll, functions in user32.dll call functions in kernel32.dll. Five of the modules shown execute in user mode (user32.dll, advapi32.dll, gdi32.dll, kernel32.dll, and ntdll.dll), and three modules execute in kernel mode (ntoskrnl.exe, win32k.sys, and hall.dll). The kernel-mode module called hall.dll provides an abstraction of the underlying hardware to the rest of the operating system. Not shown are large numbers of other interdependent modules of the operating system that do not expose APIs, but which provide critical operating system functionality. These interdependencies, or system programming interfaces, are created to allow the modular development of the operating system, not to provide application programming interfaces, hidden or otherwise.

55. All of the user-mode modules depicted in Figure 1 expose APIs that can be invoked by application programs. Most application programs requiring code to execute in kernel mode ultimately invoke code in ntdll.dll that, in turn, invokes code in ntoskrnl.exe through a special software interrupt, but, for performance reasons, the graphics subsystem is an exception to this rule. The user-mode part of the GDI (graphics display system) in gdi32.dll makes direct calls to the kernel-mode part of the GDI (found in win32k.sys). The kernel-mode part of the GDI is also called directly by code in user32.dll. User-level communications software behaves in a similar manner.

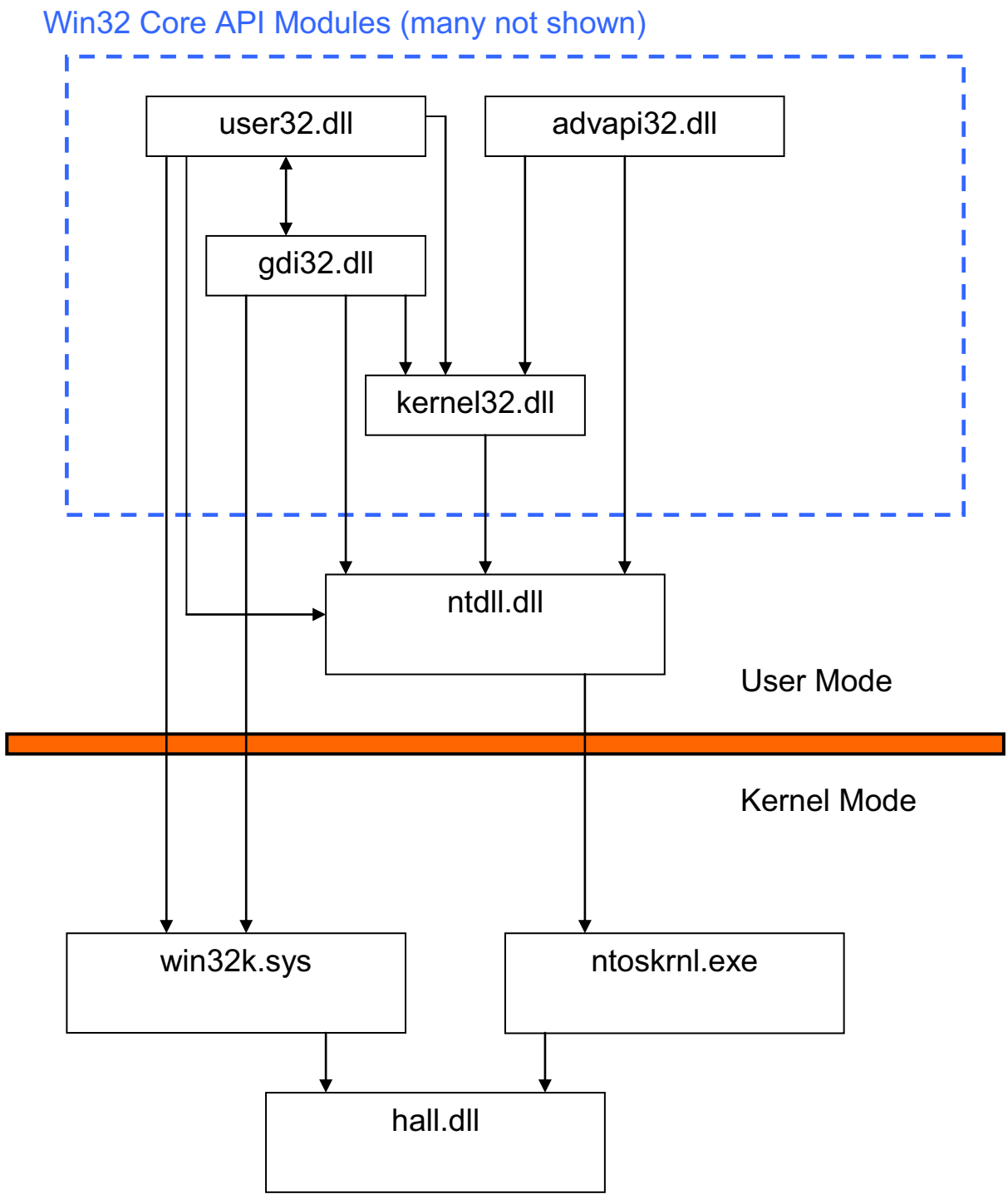


Figure 1: Some Windows OS Modules That Provide APIs

5.A.5. Middleware

56. In his report, Mr. Alepin quotes the *Microsoft Computer Dictionary*, which provides three definitions of "middleware" (Alepin Report at 25-26). In my experience the most common and sensible definition of middleware is *software that exposes a sufficiently diverse set of APIs so as to permit software designers to develop general purpose applications* (this is somewhat a blend of the first and second *Microsoft Computer Dictionary* definitions), however, as I describe below, certain terms relevant to this case have been precisely defined during previous litigation.

57. Mr. Alepin states in his report that "*An important commercial aspect of middleware is that it is often 'cross-platform,' meaning it 'can function on a variety of computer hardware and operating systems'*." (Alepin Report at 26). Mr. Alepin's arbitrary requirement that in order for software to be "middleware" it must be cross-platform is not in keeping with any common usage of the term "middleware" that I have encountered in my career, and I can see no technical justification for that limitation. Indeed, the great majority of middleware of which I am aware is not cross-platform. A defining aspect of middleware is that it provides a simplifying abstraction of lower-level operating system software. For example, the Microsoft Foundation Classes (MFC) provide a simplifying abstraction of the Win32 API set. The MFC library makes it easier to develop applications with rich functionality quickly. However, middleware as a simplifying abstraction typically does not expose the full power of the underlying interfaces.

58. Mr. Alepin asserts in his report that he considers office productivity software (including groupware), as well as email messaging software, to be "middleware", *e.g.*, "*PerfectOffice 3.0 was a form of middleware because it exposed APIs that could enable software developers to*

write programs to the suite" (Alepin Report at 11); "*Office productivity application suites including groupware also can serve as middleware platforms, by providing APIs and scripting capabilities for third-party developers*" (Alepin Report at 47); and "*MAPI 1.0 refers to the entire MAPI architecture, including the client APIs, Simple and Extended MAPI, the SPI, and the MAPI subsystem, which is a form of middleware that Microsoft bundled with the operating system*" (Alepin Report at 114). The fact that a piece of software exposes APIs for use by other software does not make that software "middleware," either as that term in commonly understood by computer scientists, or as that term has been precisely defined with respect to Microsoft Middleware in the Washington DC case (see Section 6.A.5 below). It makes no sense to me to suggest that an office productivity application, or even a suite of such applications, can somehow be considered middleware, or as a platform for general purpose applications.

59. While the APIs exposed by office productivity suites such as Microsoft Office or Novell/WordPerfect PerfectOffice gave developers the ability to perform simple actions, such as manipulating document objects, and to create simple GUI elements such as dialog boxes, these APIs did not represent the kind of development environment or tools necessary to create general purpose applications.

5.A.6. Operating System Graphical User Interface

60. Virtually all operating systems require some form of user interface. The user interface provides the means by which the user invokes operating system functionality. Prior to the advent of graphical display devices, operating system user interfaces were text based. If a user wanted to print a file, he or she would have to type a command of the form print filename. Thus, the

ability to effectively use operating system resources was constrained by the user's knowledge of the commands that accessed these resources. In many cases, commands to the operating system required the user to provide a set of arguments in a particular sequence and format. For example the Unix command awk '{print ; if (NF != 0) print ""}' single-spaced-file-name > double-spaced-file-name turns a single-spaced document into a double-spaced document. In many cases, the effective use of a text-based operating system user interface required expert knowledge of the commands associated with this interface.

61. Researchers in industry and academia began experimenting with operating system graphical user interfaces in the mid 1970's. Initially, graphical display devices were quite expensive, but this cost dropped over time, in large part due to the increasing availability of inexpensive display memory components. Operating systems with graphical user interfaces facilitate computer use by a broader population of users because users are able to cause the operating system to perform complex actions on their behalf by "pointing", "clicking", "dragging" and "dropping".

62. In his report, Mr. Alepin at times uses the terms "graphical user interface" (GUI) and "shell" interchangeably. This is consistent with common usage, but it is important, in my view, to remember that the operating system shell has been an integral part of Windows since at least the development of Windows 3.x, and was a critical component of Windows 95. In my view it makes little sense to say that the user interface is not part of the operating system, any more than it makes sense to say that the dashboard is not part of the automobile. The Microsoft definition of "GUI" quoted by Mr. Alepin on pages 23-24 of his report focuses on the GUIs of applications, not of operating systems.

63. The integration of the GUI into the operating systems ensures that all applications benefit from this functionality and that the user has a common experience across multiple applications and the operating system. For example, application developers use this common functionality to provide user interface elements such as pull-down menus and dialog boxes within their applications. Applications that make use of this common functionality therefore offer to users dialog boxes and pull down menus that look and behave in familiar ways across multiple applications. This common functionality also allows the user to interact with both applications and the operating system in familiar ways to browse the file directory, "drag and drop", "cut and paste", and perform similar operations. This is a significant benefit to all concerned. If the GUI were not a part of the operating system, application software developers would have to:

- Create their own GUI for each application, which would likely behave differently than the GUI of another vendor's GUI (and the operating system), or
- Require that users purchase some (perhaps specific) third-party GUI product (in addition to the operating system and the application).

64. Option 1 is likely to adversely impact development time for the application, and adversely impact users to the extent that different applications (and the operating system) have a different "look and feel" and diverging methods for operating their GUI. Option 2, in addition to likely costing the user more money, creates a substantial support problem for both the application developer and the user, since in this case the application and the third-party GUI must be kept current with compatible upgrades. Integration of the GUI into the operating system, as was done with Smalltalk, Windows and the Mac OS (among others), avoids these serious pitfalls.

5.A.7. Integration of Functionality Into the Operating System

65. In general, incorporation of features or functionality into the operating system from software products that were previously provided separately has numerous well-known technical benefits, in addition to the obvious benefits to the user. For example, when DOS and Windows were integrated into Windows 95, the new product provided a single install, reduced testing burdens, plug and play capability, and the ability to use long file names. The same is true of the integration of web browsing capability into the operating system – there were certain technical benefits that could not be achieved when the functionality was made available separately. In my view, such integration of functionality represents the natural evolution of operating systems produced by Microsoft, and all other vendors of general purpose operating systems.

6. Allegations Raised in the Report of Plaintiff's Expert

6.A. General Observations

6.A.1. Lack of Basis, Foundation, Meaningful Citation in Alepin Report

66. Mr. Alepin's Report of May 2009 contains many statements and opinions for which no basis or foundation is apparent, for which little specificity is provided, or for which no meaningful citation is provided. Examples of such statements and opinions in Mr. Alepin's report include the following (the page number of the citation to Mr. Alepin's report is shown in parentheses):

- *There was no technical justification for de-documenting the namespace extensions. Microsoft's contemporaneous and post hoc rationalizations for the decision are unsupportable and inconsistent with Gates' written decision. (13)*

- *Based on my review of the record and my knowledge of software development, I conclude that Novell's original plan to release a suite optimized for Windows 95 by September 1995 was reasonable and would likely have been attained but for Microsoft's decision to de-document the namespace extensions. (14)*
- *Bundling its email client with Windows 95 gave Microsoft a technical and distribution advantage over rivals. (16)*
- *Based on my review of the record and discussions with former developers, I believe that Novell would have likely released its Windows 95 suite close to its September 1995 target date but for Microsoft's conduct. (18)*
- *Purely "internal" APIs are distinguished from other APIs by virtue of their use by a prominent application or piece of middleware. (32)*
- *As part of its long-running effort to drive Novell off the desktop, Microsoft took steps to leverage its desktop operating system monopoly to block Novell from gaining access to corporate network customers. Microsoft bundled its own client-side networking software to ensure that it - not Novell - was the provider of software used to connect Windows PCs to the network. (44)*
- *Microsoft's inclusion of client-side networking software, especially the NetWare client, adversely affected Novell and its NetWare products. In essence, Microsoft sought to ensure that it was the provider of the software used to attach Windows-based PCs to any local area network. (44-45)*
- *Novell's historic middleware threat to Microsoft increased when Novell acquired WordPerfect and its evolving PerfectOffice and GroupWise platforms. (57)*
- *In my opinion based on my technical and industry experience, GroupWise posed a middleware threat to Microsoft's operating systems. (66)*
- *Microsoft refused to participate in the OpenDoc consortium or to support OpenDoc in its applications, and eventually denied CIL members access to information about Chicago, making it very difficult for CIL to make OpenDoc available for Windows 95. This had profound consequences for applications developers, the CIL members, and the platforms supported by OpenDoc. (70)*
- *Microsoft claimed that it was invited to join only at the last minute. OpenDoc was not the only middleware industry standard effort that Microsoft refused to join, promoting instead its own proprietary specification. Microsoft had refused to join standards covering database access and, as discussed above, messaging (VIM). (70, Footnote 296)*
- *Just as Microsoft was not in a position in 1995 to address the paradigm shift represented by Netscape and the Internet, Microsoft was not in a position to address the paradigm shift to networking, groupware, and applications with platform-like capabilities. (76)*
- *In late September 1994, Novell was leading in networking technologies and had the leading market share. (78)*

- *In contrast to Novell, Microsoft was about to launch the Microsoft Network a closed, proprietary content and format, Windows-only online service that was to compete against AOL. According to some Microsoft testimony and pleadings in United States v. Microsoft, Microsoft executives realized in the middle of 1994 that they should have been betting on the Internet. (78, Footnote 353)*
- *Gates ordered the withdrawal of the documentation for the namespace extension mechanism, in effect preventing Novell/WordPerfect and Lotus from shipping applications that used the extensions as they were planning to do - before Microsoft Office could do so. (91)*
- *Gates wanted his Office developers to have exclusive use of the namespace extension mechanism in an Office shell - as distinct from an operating system shell - that would become "the single place where users can find and manipulate all their information irrespective of its type, including all documents and files, in addition to personal information such as appointments, task lists and mail." (91-92)*
- *Some of Microsoft's own applications and Windows components were allowed to use the namespace extensions after certain of Microsoft's teams pleaded for special treatment. (93)*
- *I believe, based on my review of the evidence, that Microsoft configured Marvel to open in a separate window in order to appear that it was not utilizing the namespace extensions, even when Marvel continued to do so, because Microsoft feared ISVs would criticize its continued use of the extensions even as it actually did so. From my review of the evidence in light of my experience in the industry, I can draw no other reasonable conclusion. (93-94)*
- *Any incompatibilities were minimal in the first instance and evaporated when Gates scrapped the Cairo shell's codebase in favor of the Chicago shell's codebase. A simple port of the interfaces to future releases of Windows on the NT architecture was now possible. (96)*
- *I recognize that all design plans are open to question and I therefore build into my opinion a two-month margin of error, meaning that there was a possibility that even absent Microsoft's conduct, Novell may not have delivered its Windows 95-optimized suite until November 1995. I do not believe that Novell would have been any later than November 1995, however. (99, Footnote 496)*
- *Based on my knowledge of software development, my review of evidence, and my conversations with former PerfectOffice developers, I do not believe that Novell could have accelerated the development of the suite simply by adding yet more developers to address the problem. To the contrary, adding more developers likely could have slowed the response time. (100)*

- *In my technical opinion, the de-documentation of the Chicago namespace extensions prevented Novell/WordPerfect from seamlessly integrating its applications, browsers, viewers, information resources, and network services into the Windows Explorer. (102)*
- *Moreover, while developers were able to implement many features for the Corel suite, these features provided less functionality when compared to Novell's plans - a loss that is attributable to Microsoft's conduct. (105)*
- *In my opinion, this loss of features and functionality is attributable to both the diversion of resources required to address and recreate the lost namespace extensions and the limitations inherent in the WordPerfect-created interfaces as compared to what existed in Windows 95. As a technical matter, the problems caused by the de-documentation prevented Novell from providing or adopting namespace navigation that was completely consistent with the Windows 95 operating system. (105)*
- *In the late 1980s and early 1990s, groupware emerged as the next "killer app" on the PC; it is a software product category that usually includes electronic mail and messaging, collaboration and file sharing, calendaring/scheduling and address book management.(105)*
- *Groupware was a threat to Microsoft. It provided application programming interfaces and behaved like a platform, attracting software developers to it. Microsoft trailed the groupware market leaders significantly, in functionality, mindshare and market share. (105)*
- *Microsoft's tactics succeeded in eliminating the groupware threat to its operating systems. (106)*
- *Microsoft's executives concluded that it would be impossible to unseat the networking incumbent without bundling groupware APIs and a groupware client with Windows 95. (106, Footnote 537)*
- *Novell, with its PC-based NetWare client, was the leader in file and print, user, and group administration servers. Its relationships with enterprise vendors, including IBM, made it a preferred partner. (110)*
- *The term "MAPI Extensions" refers to Microsoft's undisclosed changes to MAPI that prevented interoperability among third-party products. (115)*
- *All of Microsoft's MAPI extensions could have been implemented using the MAPI standard; there was no technical reason not to document the extensions as part of the standard. (115)*
- *Microsoft thus guaranteed the defeat of VIM, just as it did or would do to other standards, such as OpenDoc and VCPI. (118)*
- *There is some evidence in the record that the VIM standard was, at least initially, inferior to the MAPI standard. Any difference in quality, in my opinion, does not explain its lack of success. (118, Footnote 581)*

- *As soon as there were Microsoft products that could leverage its control over the MAPI standard, Microsoft deprived third-party products of the prospect of true interoperability and gave its own products special interfaces that uniquely empowered them when compared to those from the competition, channeling improvements through its private specifications. (120)*
- *In fact, the standard was developed from Microsoft's own product – not the other way around, meaning the standard was tuned, tweaked and resolved to allow Microsoft's products to work optimally. Every issue would be resolved using the "reference implementation" found in Microsoft's own products. (121)*
- *In the early 1990s, Microsoft undertook to create a "Notes killer" application code named Bullet. (122)*
- *This type of decision-making was repeated by Microsoft for documentation decisions, covering messaging APIs in Chicago and the related client components, Capone, Athena, etc. Moreover, Microsoft used this same logic when determining whether to document other essential interoperability interfaces in different product areas, such as Exchange and Internet Explorer. (123)*
- *MAPI was to be the crucial piece that changed the outcome for Microsoft. (124)*
- *Microsoft faced the same issue with respect to interface disclosure and bundled applications for Capone and Windows 95. (124)*
- *Third parties could see that the developers of the crucial middleware on which they would rely for the success of their product were being developed by the same people within Microsoft that worked on directly competing products to those third parties. (124, Footnote 612)*
- *In the absence of technical justification, in January 1994, Microsoft sought to recruit another operating systems vendor, Apple, that would put messaging in its operating system, thereby establishing a pattern for Microsoft's decision. (125)*
- *By persuading Apple to place messaging in the Mac OS, Microsoft was able to generate both (i) ground cover for its bundling of messaging in Windows and (ii) the ability to argue that messaging should be in the operating system - contrary to the approach Lotus and others were advocating. (125)*
- *As seen over time, when confronting middleware threats to its operating system, such as workgroup software, Internet browsers, and media players, Microsoft's response was to assert the technical inevitability of the combination of the middleware with operating systems generally, and with its operating system in particular, thereby discouraging the development and purchase of related software. (125)*
- *Microsoft's success in integrating messaging as part of the operating system would have profound effects on customers and third-party developers. For customers, their purchasing decisions would be influenced by whether the third-party products used*

operating system-based messaging. For third-party software developers, given Microsoft's position as the dominant operating system vendor and the dominant desktop applications vendor, Microsoft could make challenging operating systems-based messaging a very expensive proposition. (126)

- *Microsoft's retreat from MAPI for the Macintosh resembles its retreat from Internet Explorer for UNIX and the Macintosh. (126)*
- *Microsoft's handling of MAPI was related to its actions against Netscape's browser. (126, Footnote 625)*
- *Essentially every statement made on page 127 of the Alepin Report.* (127)
- *As with the namespace extensions, Microsoft became concerned that its messaging client software would enable companies such as Novell to deliver applications that were functionally superior to those offered by Microsoft. (128)*
- *To ensure that Microsoft's workgroup applications would always enjoy a competitive advantage, Microsoft planned to add extensions to MAPI that it would not document. In so doing, Microsoft exploited its position as the owner and controller of the MAPI standard to create extensions unique to its client and server messaging products. (129)*
- *A user of Microsoft's Exchange client (Capone) would receive 100% of the value of the software purchased only when it connected to a Microsoft Exchange server; when connected to a third-party server, some percentage of functionality was lost, even if that server was MAPI compliant. (129)*
- *Such use of secret, undocumented specifications inside a supposedly open standard deprives the standard of meaning and purpose. (129)*
- *The net result was the destruction of cross-vendor interoperability when Microsoft products were involved on one side or the other - the Exchange client worked fully only with the Exchange server. (132)*
- *Microsoft strategically limited third parties' ability to achieve interoperability with Microsoft's products through its disclosure practice. (133)*
- *Microsoft has repeatedly used this strategy to control the quality of the competition it faces. (134)*
- *Further, my technical research has verified that Microsoft's Bob used a 16-bit version of Extended MAPI. (135, Footnote 660)*
- *Third parties reasonably expected that MAPI middleware would be installed on the Windows operating systems out of the box, or in the absence of this, have a consistent mechanism for installation. Third parties also expected that the version of MAPI installed on the relevant operating system would be the same version they tested their software with during development (except, of course, for bug fixes). (136)*
- *Microsoft engaged in similar conduct with Windows 2000. (136, Footnote 662)*

- *Microsoft treated MAPI as it would a normal system file rather than an industry standard. In failing to abide by its commitment to integrate MAPI with the operating system and treat it as a standard, Microsoft ignored reasonable third-party expectations about how MAPI would be implemented. Microsoft's plan all along was to bundle its mail client with MAPI and to install the two of them together, regardless of whether the user sought to install MAPI only for use with third-party messaging clients. (137)*
- *In this way Microsoft would be in the same position as it was with Internet Explorer: able to ensure that its mail client was installed on every Windows-based messaging-capable PC in the world. (137)*
- *Third parties reasonably expected that Microsoft would not create disruptive changes to the MAPI middleware without first providing third parties notification and the opportunity to test their software to ensure proper operation. (138)*
- *Microsoft did not properly consider third parties' reliance on the MAPI middleware. (138)*
- *In my opinion and based on my experience, it is virtually certain that these third parties, with proper notice, would have told their customers of the potential for conflicts when installing Office 97 and would have tested their products with the new DLL to avoid harming mutual customers. (139-140)*
- *Customers who chose Novell or other third-party email software to work with Microsoft's messaging software could fail at any moment - Microsoft could and would cause such failures. (141)*
- *Given Microsoft's apparent lack of concern, Microsoft's ability to alter MAPI and its deployment could simply break the version of MAPI a third party had rights to distribute. In fact, the usefulness of redistribution rights given by Microsoft turned out to be quite ephemeral. (141-142)*
- *After the VIM consortium agreed to support MAPI and drop its own standard, Microsoft dropped all pretense of vendor neutrality. (142)*
- *Microsoft also had an obligation as the MAPI middleware supplier to test its products with third-party products, or to provide to third parties notice of any changes and an opportunity to test against those changes. (143)*
- *After the first wave of MAPI 1.0 products reached the market, Lotus and Novell saw the need to expand the functions covered by the MAPI standard. Both proposed changes to Microsoft, though such overtures were declined. After having moved its competitors to its own specification, Microsoft toned down its MAPI evangelism, and its conduct signaled to the industry a de-commitment from MAPI as a standard. (143)*
- *Further, having beaten Netscape, Microsoft could not foresee the emergence of any competing, open internet-based standards for groupware messaging. Microsoft could finally return to the MAPI specification, proprietarily extending it for the exclusive*

benefit of Exchange and Outlook, without needing to allow third parties to participate. Microsoft disregarded third parties relying on MAPI when it de-committed, effectively depriving those parties of any means to evolve the standard. (143)

- *I analyzed both Novell's original analysis and Microsoft's reply and, in my technical opinion, have concluded that Novell's assessment of the reasons for its inability to comply with the logo requirement was correct. Architectural differences between Windows 95 and Windows NT - coupled with ambiguities as to whether the application needed to be compatible with Windows NT 3.5 or 3.5.1 - were the root cause of Novell's issues. (153)*
- *WordPerfect's applications were frequently cited for their superior formatting and printing functions, and WordPerfect was committed to providing superior functions on Windows 95. (155)*
- *Combining Qcodes with background printing in Windows 95 meant that Novell/WordPerfect could achieve superior performance when compared to other word processors on Windows 95. (158)*
- *Novell's central problem in developing PFPS concerned documented APIs that had no corresponding functionality. The APIs were "stubbed out" with Microsoft's promise of future delivery of the expected functionality. The functionality would have allowed Novell's applications to set a data type for Qcode and implement a print processor that could interpret Qcodes; without the promised functionality, Novell could not utilize the Windows 95 printing system as it was led to believe. (160)*
- *Relying on Microsoft's commitment to deliver the ability to set custom data types in time for the initial release of Windows 95, Novell undertook the development of PFPS. (160)*
- *Novell contacted Microsoft in the early part of 1995 and was assured that its development plans were feasible and consistent with the printing functionality that Windows 95 would deliver. (160-161)*
- *Consequently, Novell's plans were not realizable. (162)*
- *Microsoft's failure to provide this functionality in the operating system did in fact cost Novell substantial resources and degraded its final products' functionality. (163)*
- *After repeated assurances extending over a period of time at the end of which Novell would have no alternatives, Microsoft informed Novell that it was reneging on its commitment to provide the needed functionality. Microsoft's own applications and its emergent file and print server were unaffected by its failure to deliver on its commitment, and indeed were beneficiaries of not having to compete against higher quality products. By refusing to make this printing functionality available on Windows 95, while it was already available on Windows NT, Microsoft guaranteed itself that it would not have to compete with a more capable product, especially in enterprise environments. Microsoft's decision prevented Novell from taking full advantage of the Windows 95 printing subsystem, and severely degraded both the actual and planned functionality of PerfectOffice. Microsoft's notice, coming two weeks prior to the cutting of the Windows*

95 gold master, made it impossible for Novell to incorporate its PFPS into the Windows 95 printing system in time for the operating system's launch. In fact, Microsoft's notice would have resulted in a substantial delay well beyond the launch of Windows 95. (163)

- *Had Microsoft not evangelized, repeatedly committed and then declined to provide the needed functionality, Novell could have saved the substantial time and resources it took to end up with a lackluster solution that alienated its customers. (164)*
- *Given that Microsoft had already implemented the particular functionality in its Windows NT system, Microsoft could have provided the promised functionality in the time frame needed by Novell's developers. (164)*
- *Following the release of WordPerfect 6.1, Microsoft documents reflect an awareness of Novell's ability to deliver highly competitive Windows applications. Microsoft readily could have appreciated the additional benefits Novell's products could have accrued from superior printing. Similarly, Microsoft could have understood the technical and business import of refusing to provide Novell with these printing interfaces. As a software development company, Microsoft understood the deleterious effect of delaying communication of its intention not to provide the required functionality until the very last moment. From a technical perspective, Microsoft's conduct is similar to its conduct with respect to the namespace extensions. (164)*

67. These assertions do not appear to be based upon any identifiable technical assessment, nor do they appear to reflect technical objectivity. Until such time as Mr. Alepin provides basis, foundation, specificity or meaningful citation for these and similar statements, I am unable to respond fully to his stated opinions and conclusions. I reserve the right to respond if and when Mr. Alepin provides some basis for his assertions.

6.A.2. Conflation of Unrelated Issues in Alepin Report

68. Mr. Alepin's report contains several statements of the form "*Microsoft's conduct <with respect to Subject Matter A in Case B> was similar, <from a technical standpoint>, to "Microsoft's conduct <with respect to Subject Matter C in Case D>*." Examples of such

comparisons include the following (the page number of the citation to Mr. Alepin's report is shown in parentheses):

- *Microsoft responded to the Novell/WordPerfect threat with a course of conduct that, in my opinion, based on my technical and business experience in the industry, (a) severely compromised Novell's ability to develop and market products with alternative technology and features to Microsoft's PC products, including Windows; (b) unnecessarily delayed and reduced the functionality of key Novell's products; (c) was similar, from a technical standpoint, to Microsoft's conduct alleged and/or found to be anticompetitive in other cases; (d) was not technically justified; and (e) impeded technological innovation in the software industry. (11)*
- *Bundling its email client with Windows 95 gave Microsoft a technical and distribution advantage over rivals - similar to the way that Microsoft's integration of Internet Explorer into Windows gave it significant advantages. (16)*
- *Microsoft's conduct towards Novell/WordPerfect was similar, from a technical standpoint, to conduct alleged and/or found in other lawsuits against Microsoft to be anticompetitive. (17)*
- *Microsoft sought to eliminate the threat posed by this new category of software through a series of tactics that are strikingly similar to those it employed against Netscape and DR-DOS. (105)*
- *Microsoft's conduct here, with respect to the bundling of the messaging middleware, raises issues similar to the browser issues raised in United States v. Microsoft. (125-126)*
- *This is similar to the Java pollution discussed in Judge Jackson's Findings of Fact. (130)*
- *Microsoft engaged in similar conduct with Windows 2000. (136)*
- *From a technical perspective, Microsoft's conduct is similar to its conduct with respect to the namespace extensions. (164)*
- *Microsoft's conduct towards Novell/WordPerfect (described above) was similar, from a technical standpoint, to conduct alleged and/or found in other lawsuits against Microsoft to be anticompetitive. (165)*
- *And similar to its conduct with respect to namespace extensions, Microsoft repeatedly used its control over Windows interfaces to advantage technically its own applications development efforts, either by withholding documentation of interfaces from competitors altogether or disclosing them only after Microsoft's applications developers implemented them. (165)*

69. Mr. Alepin offers little or no analysis to support such statements, and it is difficult for me to imagine the relevance or benefit of such comparisons. These statements appear to me to represent biased advocacy based upon asserted patterns of behavior, rather than technical analysis of the allegations made by Novell in this case. Should Mr. Alepin later provide some analysis and discussion of the relevance he sees in these comparisons, I may respond.

6.A.3. Mr. Alepin's Reliance on Undocumented Interviews

70. Mr. Alepin offers as the basis for some of his opinions interviews that he conducted with various individuals. Mr. Alepin has not produced transcripts (or in fact any record) of these interviews. I am unable to respond fully to Mr. Alepin's opinions based upon these interviews until such time as I have access to these same individuals or to accurate records of the questions asked and the answers they provided during their interviews.

6.A.4. Self Citation in Alepin Report

71. In some instances, Mr. Alepin cites to his reports in a prior case as the basis for his opinions in this case. To the extent that these citations are to conclusions previously offered with insufficient basis, I am unable to respond fully until such time as Mr. Alepin provides meaningful support for the conclusions he seeks to import into this case.

6.A.5. The Need for Precision When Defining Terms

72. Mr. Alepin's report contains a variety of definitions. In many cases these definitions are drawn from general purpose dictionaries, such as the *Microsoft Press Computer Dictionary*, or the *Barron's Dictionary of Computer and Internet Terms*, and in some cases they have been

created by Mr. Alepin. It is normal for expert witnesses to provide or cite definitions for terms of art in an effort to explain complex matters to the Jury. However, this case requires care in defining terms, as I describe below.

73. Many of the definitions of importance in this case should be understood in light of the legal determinations that have been made with respect to those definitions. In particular, the Final Judgment that resulted from the Washington DC case contains precise definitions for terms such as "API", "Communications Protocol", "Documentation", "Microsoft Middleware", "Microsoft Middleware Product", "Operating System", and "Personal Computer", among others. Because these definitions govern Microsoft's responsibilities under the Final Judgment, it makes sense to me that they should be considered in discussing claims that Microsoft engaged in anticompetitive conduct in the PC operating systems business.

74. General purpose dictionaries such as the *Microsoft Computer Dictionary* are intended to help ordinary computer users gain a basic understanding of certain principles of computer science and engineering. Such general purpose dictionaries are not a guide to operating system design. Thus, reliance on general purpose dictionaries, which necessarily contain highly simplified definitions, may lead to erroneous conclusions.

75. In a particularly relevant example, Mr. Alepin appears to have an overly broad view of the meaning of application programming interfaces. This view is reflected in much of Mr. Alepin's discussion of this topic. Application Programming Interfaces (APIs), as their name implies, consist of those interfaces used by applications to invoke operating system functionality. As I have previously noted, interfaces provided by operating system components to other operating

system components are correctly termed system programming interfaces, not application programming interfaces.

76. The fact that a piece of software exposes APIs does not necessarily make that piece of software a "platform", or even "middleware", as those terms are commonly used. Absent litigated definitions, these terms are typically reserved for those pieces of software that seek to expose a set of APIs sufficient to develop general purpose applications (*e.g.*, the Microsoft Foundation Classes), as opposed to narrow, special purpose applications (*e.g.*, a media player). It is perhaps appropriate to refer generally to a media player that exposes APIs as "special-purpose middleware" (as distinguished from "general-purpose middleware" such as the MFC), but in my view, and as I discuss below, neither office productivity suites nor MAPI software represent any kind of middleware.

6.B. Allegations Regarding "Shell Namespace Extensions"

77. In his May 2009 report, Mr. Alepin alleges that "*Microsoft's decision to remove the namespace extension application programming interfaces ('APIs') from Windows 95 was not technically justified, particularly given its prior representations to developers through and after the first Windows 95 beta release that it would support developers' use of those APIs*." (Alepin Report at 6). Elsewhere in his report, Mr. Alepin alleges that Microsoft "*withdrew support for*" (or "*de-documented*") the shell namespace extensions (Alepin Report at 90 and 98). Despite this internal inconsistency, it is clear (and discussed more fully below) that the software code underlying the shell namespace extension functionality remained in Windows 95, but was not formally documented before the commercial release of Windows 95.

78. Shell namespace extensions represented functionality that permitted the Windows Explorer feature of Windows 95 to represent hierarchical data in a folder-like metaphor. This so-called "tree view" appeared in the left-hand pane of the Windows Explorer, also referred to in Microsoft documents as the "scope pane." This "tree view" allowed software developers to represent, and users to view, data in a manner consistent with the way in which users viewed files in a directory. This functionality was introduced and documented in the first "beta" release of Windows 95 (sometimes referred to as the "M6" release). The shell namespace extension functionality was not removed from Windows 95, but, for reasons explained below, documentation for some of this functionality was not included in the second "beta" release of Windows 95 (sometimes referred to as the "M7" release) (Richardson 107:13-15; Maritz 140:11-15 and Ex. 22). Of course, since software developers had already received documentation of the shell namespace extension mechanism with the M6 beta, they already had the information necessary to understand and use that mechanism. As I describe below, the record reflects that software developers were promptly informed of the decision (and its rationale) not to formally document the shell namespace extension mechanism (MX 6055840). They were also given detailed documentation regarding alternative approaches to achieving equivalent functionality that were less problematic (MX 6055840-44 at 41; NOV00685571-72; NOV-B01644605-17; Struss 156:17-22; 176:19-21). In the months to follow, partial documentation for the shell namespace extension functionality (in addition to that already supplied with the M6 beta) was made available by both Microsoft and persons external to Microsoft (Belfiore Exhibit 19). Approximately a year later, after additional development to address robustness issues, the shell

namespace extension functionality was fully documented on the Microsoft Developer Network (MSDN).

79. I have reviewed carefully the substantial evidentiary record related to this matter that was developed during discovery during this case, and to some extent, in prior cases. Based upon my evaluation of the evidence, I conclude, in contrast to Mr. Alepin's views, that:

- the decision by Microsoft not to formally document some of the shell namespace extension functionality was a normal part of the software development process for large and complex products such as Windows 95;
- there were sound technical reasons for deciding not to document formally the shell namespace extensions;
- Microsoft made a concerted effort to mitigate the effect of this decision for ISVs (including Novell/WordPerfect developers);
- WordPerfect software development for Windows 95 had not actually begun when the decision not to document formally the shell namespace extensions was made; in fact, the record reflects that Novell/WordPerfect never wrote any product code in the relevant time frame that made use of the functionality in question;
- there is no record of which I am aware of a high-level (corporate officer at Novell/WordPerfect to corporate officer at Microsoft) complaint or communication regarding the shell namespace extensions, although there is evidence of many high-level complaints and communications regarding other subjects;

- at no time did Microsoft Office products (which I understand to represent Novell/WordPerfect's competition in suites of office productivity applications) ever take advantage of the shell namespace extension functionality for which formal documentation was temporarily unavailable;
- during the relevant timeframe, software developers had several options for incorporating shell namespace extensions into their products;
- there are reasons unrelated to any alleged Microsoft conduct why Novell/WordPerfect's development of 32-bit versions of its office productivity applications for Windows 95 was likely to be problematic; and
- certain deposition testimony of Adam Harral and Gregory Richardson in 2001 concerning shell namespace extensions contain statements that are inconsistent with the rest of the evidentiary record in this case.

80. Below, I discuss each of these conclusions in turn.

6.B.1. Context

81. Before discussing my conclusions, I will attempt to put these issues in context. I believe it helpful to describe various aspects of the state of affairs that existed at Microsoft in the 1993-1996 timeframe, as I have come to understand them from the record. During this period, there were two primary operating systems development groups at Microsoft: the first, headed by Brad Silverberg, was the Windows 95 (and what would become the Windows 98, and Windows Millennium Edition) group (Silverberg 10:19-11:9; Allchin 13:14-16); the second, headed by Jim Allchin, was the Windows NT 3.51 (and what would become the Windows NT 4.0 and Windows

2000) group (Allchin 11:14-17). Mr. Silverberg and Mr. Allchin both reported to Paul Maritz, who in turn reported to Bill Gates. In addition, a third Microsoft operating systems development group was working on an ambitious operating system intended to be a successor to Windows NT, codenamed "Cairo." Cairo was to have a number of advanced features, including an object-oriented shell, that have yet to appear in Microsoft's operating systems (Muglia 40:5-10). Bob Muglia, who reported to Mr. Allchin, led the Cairo group (Muglia 26:16-22).

82. The contemporaneous email record reflects that these three operating systems development groups had sharply differing views regarding certain operating system design issues (MS-PCA 1180275-85; MS 0073067-69; MS7088886-94 at 91; MS 044118-20 at 18; MS 060468-70; see also Nakajima 55:20-59:11, 98:14-22; Madigan 110:6-111:8). This is not surprising. Windows 95 had its roots in MS-DOS and Windows 3.1; it was designed to make it easier for unsophisticated users to make effective use of their personal computers. The design of Windows 95 was focused on, among other goals, reducing complexity for users (*e.g.*, creating a simple boot process, plug and play for hardware devices, and other "usability" enhancements), and providing reasonable performance on a machine with modest resources (in particular, on a machine with only four megabytes of memory (Silverberg 108:21-25; Evslin 47:5-11; Nakajima 118:16-18; Maritz 69:8-13; Belfiore 168:17-25; MS7082450-51). Robustness, *i.e.*, the ability of an operating system to run for long periods of time without experiencing a malfunction (which in my experience is a very important issue for business users) was not the primary design objective for the Windows 95 team (Silverberg 54:14-22; 78:14-25; 109:11-20). Windows 95 was therefore highly appropriate for home users.

83. In contrast, Windows NT represented a new design from the ground up; it was designed to provide a reliable and robust networked computing environment, such as might be expected in a corporate office environment. Windows NT therefore considered reliability and robustness to be more important goals than usability and machine resource usage (Muglia 37:24-38:4). The Cairo design looked farther into the future than Windows NT, and therefore planned to take advantage of anticipated increases in microprocessor performance and memory size (Muglia 26:25-27:7). These sets of operating system design goals, while appropriate for their particular domains, were necessarily in tension when differences had to be worked out among the design groups in order to achieve shared objectives.

84. Mr. Gates had a particular vision for how future Windows operating system shells might operate; this vision was never realized (Gates 186:15-22). In fact, Mr. Gates viewed the limited functionality offered by the Windows 95 shell namespace extensions (the ability to add custom folders to the tree view in Windows Explorer) as "trivial and unimportant" (Gates 259:18-22).

85. As personal computers became more powerful, the distinction between "low-end" computing (Windows 95 users) and "high-end" computing (Windows NT, and later, Cairo, users) was becoming less distinct. The three operating systems development groups generally worked independently, but as their target audiences blurred, it appears that it became more necessary for the groups to work with one another (Muglia 31:9-10, 237:16-21; Madigan 22:6-13).

86. During the development of Windows 95, a new shell was developed that differed substantially from both Windows 3.1 and Windows NT 3.5.1 (which had similar shells, from the

user's perspective). This shell had a set of new interfaces that allowed "extension," the ability to add certain components behaviors, and the ability to extend the "namespace" (the enumeration of system resources that can be represented as files or file-like objects) (Belfiore 192:13-194:13). The operating system component that provides this enumeration is often called the "Windows Explorer." Sometimes the terms "shell" and "explorer" are used interchangeably in the documents that I have seen, but, as I noted previously, the term "shell" is sometimes used to refer to the entire GUI of the operating system, including the Start menu, the task bar, etc. Although it possessed potentially interesting features, the shell namespace extension mechanism in Windows 95 was problematic. For example, the Windows 95 shell namespace extension mechanism was incompatible with Windows NT, because shell namespace extensions in Windows 95 run in the same process space as the Windows 95 GUI (Nakajima 84:4-8; Muglia 241:17-23). Therefore an error in a shell namespace extension could crash Windows 95 (requiring a reboot) (Henson 67:6-9). Using the same process for the GUI and for shell namespace extensions was perhaps an appropriate design choice for Windows 95, the design of which valued usability over avoiding the risk of malfunctions that might be caused by misbehaved applications, but was in conflict with the robustness requirements of Windows NT (Henrich 56:12-20). Even for Windows 95, the prospect of having third-party shell namespace extensions, except under very controlled conditions, represented a serious reliability exposure (Henrich 56:21-57:4; Muglia 240:23-243:15).

87. In addition to core operating system development, Microsoft personnel were also developing other components of Windows 95, as well as applications intended to run on Windows 95. Three such software projects mentioned in the documents related to this issue are

(1) Marvel, the codename for the Microsoft Online Network (MSN) client; (2) Ren, the codename for what would become Microsoft Outlook, the email and collaboration client designed for use with Microsoft Exchange; and (3) Capone, the codename for the basic Microsoft Mail utility that was a component of Windows 95. During Windows 95 development, these three design groups apparently evaluated the use of the new shell namespace extensions (MX 6109491 at 91; MS7082580-82; MS-PCA 1384999-5010 at 003; MX 5049798-99; MX 6109491 at 91; MX 6109491).[2]

88. It was in this context that various groups and individuals at Microsoft debated the internal and external use and deployment of shell namespace extensions in Windows 95 (MS 5033031-33; MS 5042220-22). Ultimately the matter was brought to Bill Gates, who made a decision in October 1994 not to document formally certain shell namespace extension functionality in the second beta release of Windows 95 (MS-PCA 1293569-70). The following discussion summarizes my analysis related to that decision.

6.B.2. Change is a Normal and Expected Part of the Software Development Process

89. The decision not to document formally some of the shell namespace extension functionality by Microsoft was a normal part of the software development process. My understanding of the Windows 95 beta license agreement (actually titled a "non-disclosure agreement") is based upon the plain English meaning of its clear language: "**The PRODUCT may not operate correctly**

[2] The Internet Explorer 3 Internet News and Internet Mail client, codenamed Athena, may have made use of the shell namespace extension functionality, although Mr. Alepin does not devote much attention to Athena in his report. Athena was not released until after the shell namespace extensions were formally documented on MSDN.

and may be substantially modified prior to first commercial shipment." This contractual disclaimer directly contradicts Mr. Alepin's assertion that Microsoft "promised to provide in Windows 95" shell namespace functionality (Alepin Report at 11-12). In addition, such disclaimers are common in beta license agreements I have signed throughout my career. I have seen copies of the Windows 95 beta license agreement signed by representatives of both WordPerfect and Novell (MSEXP243 and MSEXP244, respectively). Further, my experience as a beta tester of many software products, including several versions of Windows and various commercial UNIX operating systems, leads me to conclude that it is normal for functionality to come and go during the beta process, even late in that process. For example, Mr. Gates observed that, although it was "very late in the day to be making changes" (Gates 252:24-25), he made a "tough decision" to "not publish the extensions" (Gates 253:25-254:1) because "what was done was so trivial that I don't -- I didn't think it was worth the trouble and I didn't think it would affect Chicago and I didn't think it would affect Capone" (Gates 255:5-9) and "before we try and do anything like this again, let's do it in a way that would allow some real integration" (Gates 255:16-18).

6.B.3. Technical Basis for Not Formally Documenting Shell Namespace Extensions

90. There were sound technical reasons for deciding not to document formally certain shell namespace extension functionality, because, as I noted previously, shell namespace extensions were implemented in the same process as the remainder of the Windows 95 GUI (Nakajima 84:4-8; Muglia 241:17-23). This limitation of the Windows 95 software architecture meant that a software bug in an ISV's shell namespace extension could render the operating system

unusable, a highly undesirable result that could result in the loss of data and a need to reboot the machine (Henrich 56:21-57:4; Muglia 240:23-243:15; MSC 0080572). Therefore, the conservative choice from a system integrity perspective was to only allow "trusted" software, whose behavior was both predictable and tested, to use the shell namespace extensions.

91. The ongoing debate among the various operating system design groups at Microsoft regarding the implementation of the shell namespace extensions also cast doubt on their future (see, for example, Nakajima 55:20-60:5; Muglia 165:10-16). In particular, the Cairo group did not want to support the Windows 95 shell namespace extensions, and had a very different vision about the way in which applications should interact with the Windows Explorer (Muglia 40:5-10; 162:13-163:6). As Mr. Gates testified at his recent deposition, the idea was not merely to add custom folders to the scope pane of the Windows Explorer for files related to a given application, but rather to use Object Linking and Embedding (OLE) technology to the application run in the right-hand pane (sometimes called the "view pane") of the Windows Explorer.

92. Once exposed in the release version of Windows 95, Microsoft would have to continue to support shell namespace extension functionality for the foreseeable future (Gates 288:11-21; C. Myhrvold 48:7-12; Silverberg at 63:14-23; Muglia 162:13-163:6). This requirement could limit future design choices for shell architecture and implementation for years into the future. The conservative choice in that situation was not to formally document the problematic (and not fully realized (Gates 186:15-22)) interfaces until such time as they were either modified to make them less problematic, or until a decision was made to remove them entirely.

6.B.4. Efforts to Mitigate the Effect of Decision on ISVs

93. Microsoft made a concerted effort to mitigate the effect on ISVs (including WordPerfect) of the decision not to document formally certain of the shell namespace extension functionality. Contemporaneous documents indicate that Microsoft (1) called ISVs as soon as the decision not to document formally the shell namespace APIs was made ("…we're now in the process of proactively notifying ISVs about the namespace api changes … So far Stac, Lotus, WP, Oracle, SCC appear to be OK with this…") (MSC 00800569), (2) that an effort was made to provide equivalent, or at least very similar, functionality ("We will provide common controls (listview, treeview, column heading, etc.) that will allow ISVs to create their own views in the same manner that the explorer does") (MSC 00800572), (3) that the details of the use of shell namespace extensions were made available to software developers by third parties (see, for example, Belfiore Exhibit 19), and (4) Microsoft addressed the same-process limitations of Windows 95 shell namespace extensions by requiring that its own use of these extensions (in Capone and Marvel) open in a separate top-level window (and therefore a separate process) (MSC 00298236). The record also reflects that Capone software was modified not to make use of the functionality in question (MS-PCA 1293571). This means that the MSN client, which was a trusted component of Windows 95, is the only arguably separate product – as opposed to features of the Windows Explorer such as My Briefcase and My Network Neighborhood – that used the shell namespace extensions that were not formally documented before the release of Windows 95.

6.B.5. WordPerfect Coding for Windows 95 had not Started

94. WordPerfect software development for Windows 95 had not actually begun when the decision was made by Microsoft not to document certain of the shell namespace extensions; in fact, as I describe below, the record reflects that WordPerfect never wrote any product code in the relevant time frame that used the functionality in question. In his report, Mr. Alepin states "*Based on my review of the record and my knowledge of software development, I conclude that Novell's original plan to release a suite optimized for Windows 95 by September 1995 was reasonable and would likely have been attained but for Microsoft's decision to de-document the namespace extensions*" (Alepin Report at 14). This statement is not supported by the evidence that I have reviewed. That record indicates that:

- WordPerfect was notified of the decision not to document formally certain of the shell namespace extension functionality in October 1994 (MSC 00800569).
- WordPerfect had reported to the Development Relations Group at Microsoft in September 1994 that "they have not begun any work on" the shell namespace extensions (MSC 00718794).
- Sid Cragun, the WordPerfect software developer given the task of considering the topic of Windows 95 Shell Integration (NOV-B01468558) makes no mention of any software development using shell namespace extensions in eighteen status reports dated February 7, 1994 through November 14, 1994 (NOV-B01468862 – NOV-B13649427).
- In Revision 1.0 of the Thunder (the WordPerfect codename for WordPerfect for Windows 95) *Concept Design Specification for Windows Shell Integration* (NOV-

800941834 - NOV-800941863), authored by Sid Cragun and dated February 21, 1995, Mr. Cragun writes:

> "Registration of custom folders which function as object containers with the same behavior as a folder. This type of shell extension is referred to as a name-space browser. To the user, it appears that the shell understands an application hierarchy that is not part of the file system. Custom folders are designed such that hierarchial [sic] relationships and contents can be displayed in the appropriate panes of the file browser window, i.e., the explorer pane and the contents pane. ***We will not take advantage of this feature since Microsoft has discontinued support of the required API since this document was originally written***." (NOV-800941844, emphasis added).

- In this document, the shell namespace extensions were at the bottom of a long list of shell integration features that Novell planned to take advantage of in Windows 95. I find it noteworthy that Mr. Cragun's statement is matter-of-fact, and does not appear to express any concern about not adding custom folders to the Windows Explorer. This is consistent with testimony from senior WordPerfect developers, who had no recollection of the shell namespace extensions being an issue in developing WordPerfect for Windows 95 (Freeman 50:1-12; Hallmeyer 47:3-20).
- Revision 1.0 of the *Concept Design Specification of PerfectFit 95 File System – File Open* (NOV-B01426168 - NOV-B01426213), dated March 31, 1995, states that "*A name-space vendor (e.g. GroupWise vends a name-space called 'In Box') can integrate a name-space with the open dialog by providing certain Component Object Model (COM) interfaces that can be called by the PerfectFit Name Space Browser code. These interfaces are enumerated below*" (NOV-B01426192). This enumeration includes IShellFolder, IContextMenu, IExtractIcon, IShellLink, ICopyHook, IFileViewerState, IFileViewer, IDataObject, IPersistFile, IPersistStream, and IPersistStorage interfaces

(NOV-B01426192 - NOV-B01426201). All of these interfaces remained documented throughout the Windows 95 development cycle, and in fact were prominently featured in a Microsoft Press book published in 1995. The book, entitled "Programmers Guide to Microsoft Windows 95"; ISBN 1-55615-834-3; QA76.76.O63P7677, is a collection of materials previously published in the Microsoft Systems Journal, Microsoft Developer Network and the MSDN Library. The subject matter in question is found in Articles 11 (Shell Namespace), 12 (Shell Extensions), and 16 (File Viewers). The level of detail and subject matter treatment is complete, and in my opinion one of ordinary skill would be able to readily put this material to use. Certainly an ISV as sophisticated as Novell/WordPerfect should have been able to make use of documented APIs to create a custom file open dialog.

- Revision 1.0 of the *Concept Design Specification of PerfectFit 95 File System – File Open* makes no mention of the shell namespace extension interfaces in question. I further note that this "Concept Design Specification" appears to me to be quite ambitious (eighty-eight different features are described) for a software component expected to be released in six months (as Mr. Alepin asserts is the case: Alepin Report at 14-15). It was, of course, Novell's choice to take on the task of developing such a complex file open dialog on a tight timetable (Ford 37:2 – 38:16).
- The "Final" "Functions and Issues" documentation for the Perfect Fit 95 Open File Dialog was released on July 13, 1995 (NOV-B02011392 - NOV-B02011406). This document states: "*This main purpose of this document is to provide a functional description of the Open Dialog for Storm. It lists function and behavior, and most*

important, a consensus of Open Dialog functionality. This document was necessary to alleviate differences of opinion of how this dialog would be implemented. ***Coding will occur from the information provided by this document***" (NOV-B02011392, emphasis added). This statement makes it clear that in July 1995, two months before Mr. Alepin's "reasonable" date of September 1995, coding of this critical component of WordPerfect for Windows 95 had not begun.

- A document entitled "PerfectFit 95 Schedule Panic Mode Modification Recommendations" (NOV-B01491962 - NOV-B01491966), dated July 28, 1995, states "*There is no conceivable way to have the NSB ['Name Space Browser (aka FileOpen Dialog'] code complete by August 22*" (NOV-B01491965). It should be noted that despite the similarity in names, the file open dialog that Novell called the name space browser was very different from using the shell namespace extensions in Windows 95 to create custom folders in Windows Explorer. The former would appear to have been a substantially more complicated undertaking than the latter, given all of the features that Novell/WordPerfect was attempting to include in its file open dialog.
- In *Novell's Objections and Responses to Microsoft's Second Set of Requests for Production*, in response to a request for WordPerfect for Windows 95 source code that used the functionality at issue, Novell responded that Novell/WordPerfect never developed any software source code using this shell namespace extension functionality:

  > "*In addition, Microsoft's decision to make IShellBrowser, IShellView, IPersistFolder, and ICommDlgBrowser 'private' and IShellFolder a 'read only public interface,' effectively prevented Novell from using the namespace extension*

> *mechanism and/or implementing the mechanism in a customized fashion.* ***Therefore, as a practical matter, no software that Novell developed could rely upon or invoke those APIs***." (Emphasis added)
>
> This statement is consistent with evidence in the record that Novell/WordPerfect had not started writing code for WordPerfect for Windows 95 until long after it was informed of Microsoft's decision not to document formally certain of the shell namespace extensions of Windows 95.

95. The record thus clearly demonstrates that even the design of WordPerfect for Windows 95 was in a state of flux until at least mid-1995; that WordPerfect was a long way from beginning to write software code when the decision was made not to document formally certain of the shell name space extension functionality; that WordPerfect was able to use, and intended to use, the available and documented functionality in Windows 95 to create its file open dialog, including the ability to display a hierarchical namespace with Novell's own custom folders; and that WordPerfect never wrote any software code in the relevant time frame that used the "de-documented" functionality in question.

6.B.6. No High-Level Complaints to Microsoft

96. There is no record of which I am aware of a high-level complaint (corporate officer at WordPerfect or Novell to corporate officer at Microsoft) regarding this matter, although there is evidence of many high-level complaints regarding other subject matters. I have examined correspondence, conference call minutes and meeting agendas/summaries between Microsoft and WordPerfect/Novell corporate leadership in the relevant time frame (late 1994 through early

1996) (NOV-B07335240-877618; NOV00707893; NOV00531888; NOV-B00882465-70; NOV-25-006858; NOV 00052662-52713; NOV00672307-705646; NOV 00513065-074). In that record I find discussion of many issues and concerns, but I find no mention of Novell's alleged concerns related to shell namespace extensions. Had these concerns been as important as Mr. Alepin alleges, I would have expected them to be raised in discussions between the two companies.

97. Mr. Gates testified that this issue was never brought to his attention, despite forums designed for the specific purpose of discussing these kinds of issues:

> Q. Mr. Struss is giving you, I guess, potential questions that might come up, and answers, in your talking points. And he says that, "The namespace extensions were initially pulled from Windows and ISVs were informed of this change. In general they've been okay with this," close quote. Do you see that, sir?
>
> **A. I see it.**
>
> Q. Did you do anything to confirm that ISVs were generally okay with this?
>
> **A. Well, I had lots of meetings with ISVs, including the one that's being referred to in the dinner here, and no one ever brought it up as an issue to me.**
>
> (Gates 296:15-297:4)
>
> Q. All right, you didn't do anything affirmatively to find out if they were okay with it; is that correct sir?
>
> **A. Well, I had dinner -- I had dinner with them and I said to them tell me anything about Windows that we're doing that you're concerned about. So there was open Q&A, huge opportunity for anybody to speak up.**
>
> (Gates 297:19-298:1)

The document referred to in Mr. Gates' deposition shows that the President, Adrian Rietveld, and the Chief Technology Officer, Dave Moon, of WordPerfect were both in attendance at the dinner in question (MSC 00696981-84 at 84).

6.B.7. Office Applications Did Not Use the Functionality in Question

98. At no time have Microsoft Office products (Word, Excel, Outlook, etc., which presumably represented WordPerfect's competition in office productivity applications) used the shell namespace extension functionality that was not documented formally for a period of time (Gates 272:18-24, 291:2-8; Belfiore 200:4-11). In fact, it was decided that this functionality even had to be removed from Capone, the simple mail client built into Windows 95, as a result of the decision not to document formally certain of the shell namespace extension functionality (MS-PCA 1293571).

99. I find this noteworthy because Mr. Alepin implies that Microsoft Office applications were somehow given an unfair advantage with respect to these interfaces (Alepin Report at 15, 81-83, 91, 95, 97). Since no Microsoft Office application, including Ren (the codename for what was to become Microsoft Outlook) made use of these interfaces (Gates 272:18-24, 291:2-8), it is difficult to see what that advantage might be.

6.B.8. Options Available to Developers

100. During the relevant timeframe, software developers had three options for incorporating the functionality in question into their products:

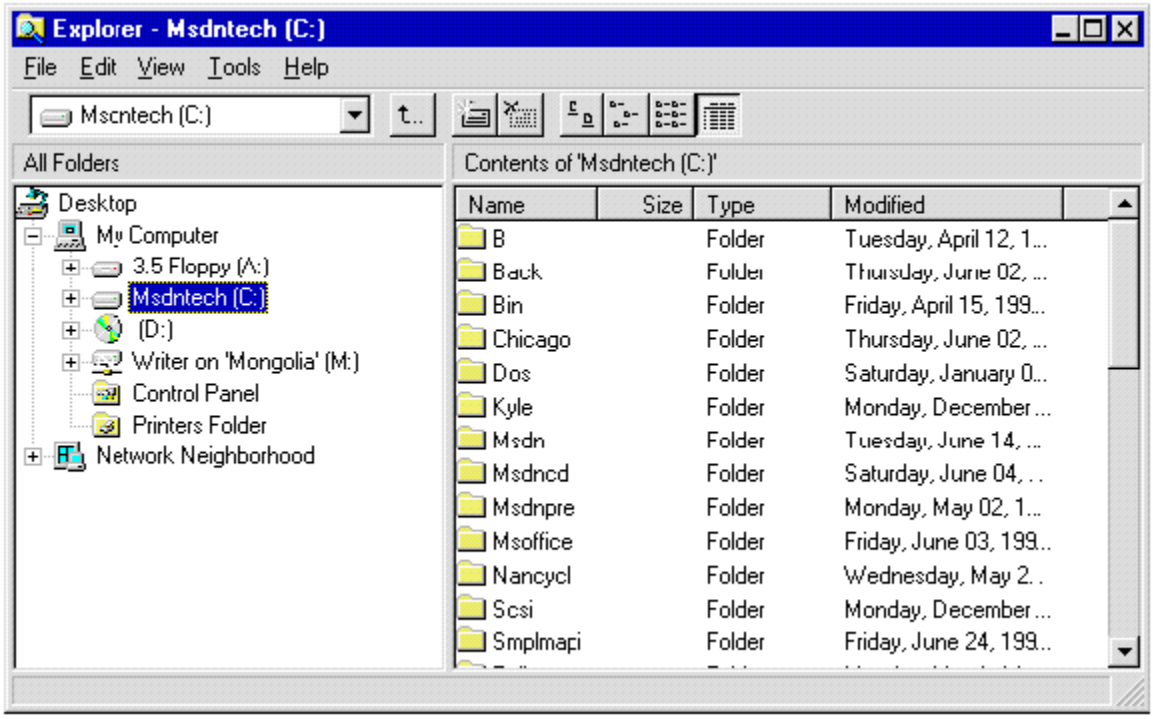
1) They could continue to use the shell namespace extension functionality that remained available in Windows 95, using the documentation provided with the first beta release of Windows 95. There was also an effort in public online forums to determine and publish

this information (Belfiore Exhibit 19). Software developers taking this course of action had to assume the risks associated with the use of undocumented operating system functionality. This is apparently the approach taken by the Marvel (MSN Client) developers (MX 5103184-85), and their use of the shell namespace extensions shows that this was not an inherently implausible course of action.

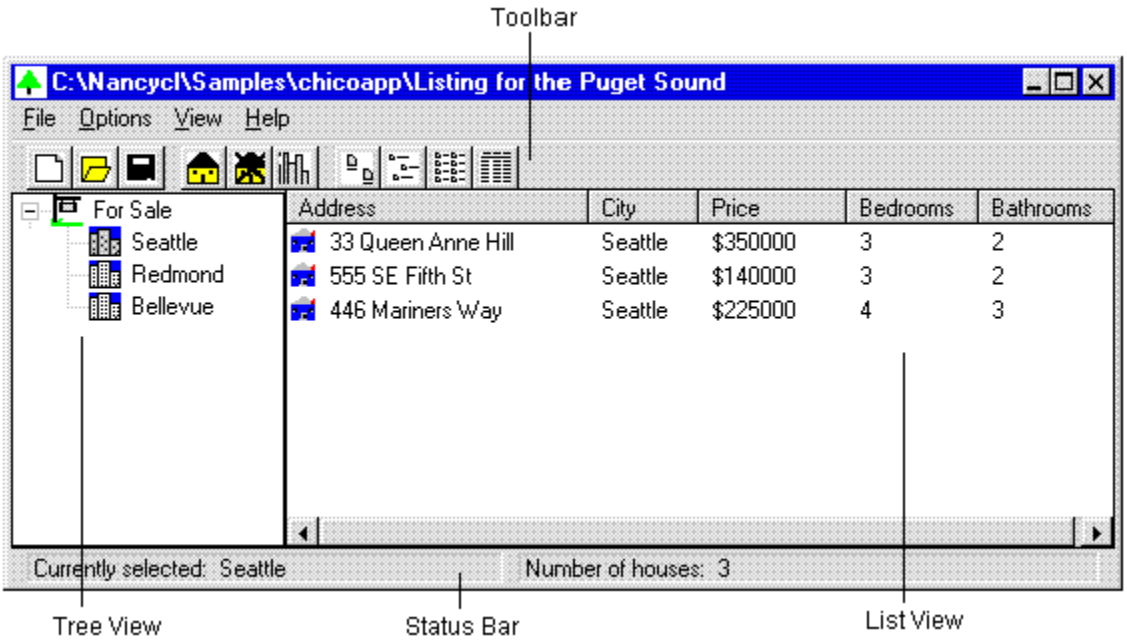
2) They could "roll their own," i.e., develop the desired user interface functionality using available and documented operating system shell interfaces. This approach, which is apparently the one taken by the Microsoft Office (including Ren), and Capone developers (Gates 272:18-24, 291:2-8) had the least risk with regard to future compatibility. As discussed above, there were a number of documented APIs in Windows 95 that Novell could – and did – use to create its own custom file open dialog.

3) They could use the "CHICOAPP" application note (first published in July 1994), entitled *Creating a Windows 95 Explorer-like Application* (http://msdn.microsoft.com/en-us/library/ms997509.aspx), and other available documentation, as a guide for how to use Windows 95 common user interface controls (listview, preview, column heading, etc.) to create equivalent visual functionality, i.e., to create views with the same look and feel as the Windows Explorer in Windows 95, but that would be compatible with the Windows 9X, Windows NT and future operating system architectures. This choice would have been an excellent choice from a future compatibility standpoint; the CHICOAPP still runs today on my Vista laptop.

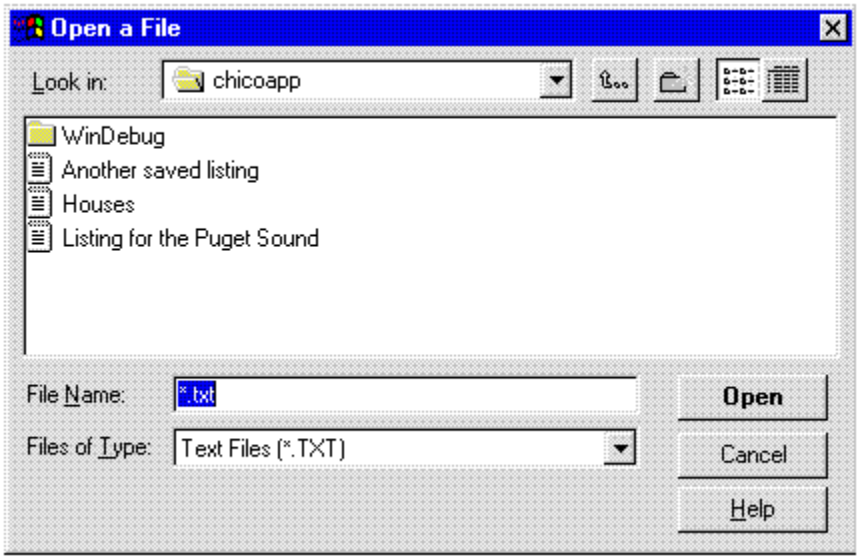
101. The CHICOAPP application demonstrated how to emulate the "look and feel" of the Windows 95 Windows Explorer. An example of the Windows 95 Windows Explorer as it appears in use is shown below. In this figure, the left pane shows a hierarchical tree view. The right pane shows the folders for the selected item in the left pane (the list view).



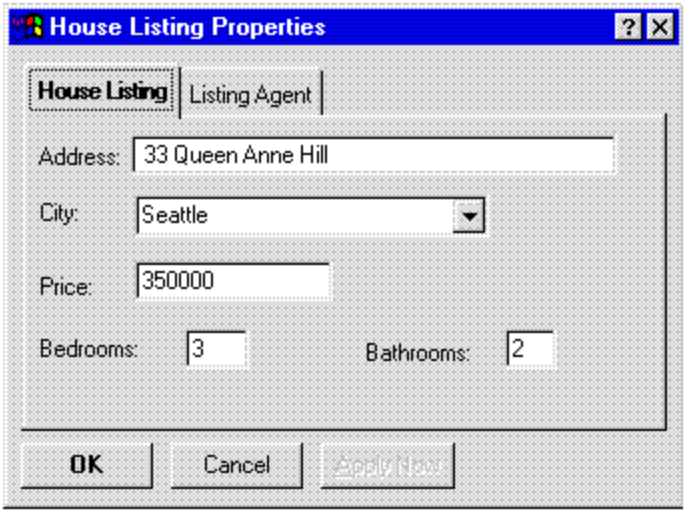
102. Shown below is the user interface window of the CHICOAPP sample application (which included full source code). As with Windows Explorer, the left pane shows a hierarchical tree view and the right pane shows the folders for the selected item in the left pane (the list view).



103. CHICOAPP also supported long filenames, and used the Windows 95 common dialogs to open and save the house-listing information. This code is similar to code written for the Windows 3.1 common dialogs. Shown below is the File Open common dialog box, which correctly handles long filenames such as “Listing for the Puget Sound” and “Another saved listing.”



104. CHICOAPP also demonstrated how to use Windows 95 property sheets. Property sheets (also known as tabbed dialogs) allow users to view and change the properties of an item. In the case of CHICOAPP, the item is a house listing. Each property sheet contains one or more overlapping windows (called *pages*) that contain a logical grouping of properties. The user switches between pages by clicking tabs that label each property page. Shown below is the House Listing property sheet page.



105. Novell clearly knew about the CHICOAPP example application, because it was produced by Novell in this case (NOV-B01644605).

106. Finally, I consider it noteworthy that these choices were the same for all parties developing new applications for Windows 95. Microsoft application developers did not take advantage of the undocumented shell namespace extensions in developing their products. Moreover, Novell developers at all times had the option of using the common dialogs in Windows 95 (including File Open), but Novell developers apparently decided that, although it "*would have been easier to use the common dialogues in Windows 95*" (Ford 37:9-12) they "*thought that they could do a better job than what was provided*" (Hallmeyer 16:4-17:3), and even though there was "*some difficulty implementing the FileOpen Dialogue because it had some features that weren't available in the Windows 95 FileOpen Dialogue*" that it was "*worth the tradeoffs*" (Hallmeyer 46:20-25).

6.B.9. WordPerfect Had Problems Unrelated to Alleged Microsoft Conduct

107. There are reasons unrelated to any alleged conduct of Microsoft why WordPerfect software development for Windows 95 was likely to be time-consuming and difficult. For example, according to the deposition testimony of senior WordPerfect for Windows 95 developers:

- WordPerfect made design choices, unrelated to any action by Microsoft, that adversely impacted both difficulty of development and delivery time of WordPerfect for Windows 95 (Ford 37:9-12; Hallmeyer 16:4-17:3; Hallmeyer 46:20-25).

- The fact that WordPerfect for Windows 95 was written in assembly language caused difficulties unrelated to any actions of Microsoft (Creighton 57:17-58:17).

- Microsoft provided support to WordPerfect software developers considered comparable to support offered to other vendors. (Creighton 84:1-85:16; 119:13-121:20; Kliger 48:4-52:19).

WordPerfect developers were apparently also hindered in their efforts by intrinsic problems with the WordPerfect for Windows software. For example, according to an internal analysis of the WordPerfect code base made by Corel after acquiring WordPerfect, the WordPerfect for Windows code base was both hard to maintain and difficult to modify in order to include new and improved features. Quoting from a Corel document entitled "Proposal for the Future of WordPerfect" (C-00453-464):

"**Hard to maintain**

- The code is extremely fragile, complex and undocumented.

- Excessive reliance on global variables means that a change in one area can cause unexpected side effects.

- Poorly conceived and designed features are essentially unfixable, for example the numbered list/outline functionality.

- Difficult for new developers to become productive. Simple tasks such as adding a command to a menu are complex and error prone.

- Significant code duplication and "similar" code (copied and tweaked) multiplies the number of bugs, but makes it difficult to be confident that one fix is appropriate everywhere.

- Uses custom-built tools for which we have no source code. For example, PIDL.EXE generates out IDL file - if Microsoft changes the IDL file format, we cannot update the utility.

Difficult to Innovate

- Difficult to add significant new features due to risks to the existing functionality. Only minor improvements are possible.

- Cannot integrate well with the other office applications – WP can't keep up with their pace of change.

- Cannot respond to Marketing/PDM/Customer requirements such as a true VBA object model, componentization, HTML 4, Cascading style sheets, SAP, Document map, Multiple page views…

- Cannot take advantage of 'new' technologies and business opportunities because the code is inflexible." (C-00456)

108. None of these many problems can be attributed to any action by Microsoft. They are all the product of the way in which WordPerfect for Windows was designed and built.

6.B.10. Harral and Richardson Deposition Testimony

109. Certain deposition testimony of Adam Harral and Gregory Richardson contains statements that are inconsistent with the other evidentiary record in this case. For example, Mr. Richardson states that Novell/WordPerfect "spent a lot of developer resources in writing specific functions for its applications to those shell extensions and APIs" (Richardson 81:6-8), but Novell internal documents do not reflect this, as described above. Both Mr. Harral and Mr. Richardson testify that Microsoft "removed" the shell namespace extension "APIs" (Harral 84:11-14 and 98:4-14; Richardson 85:4-8), but the record shows clearly that these interfaces were not removed; the only thing that happened was that Microsoft decided for a time not to document them formally, as I discuss above.

110. The testimony of Mr. Harral and Mr. Richardson also does not clearly distinguish between efforts to add custom folders to the Windows Explorer (for which using the shell namespace extensions was one of several possible options) with Novell/WordPerfect's efforts to create an advanced file open dialog that could search across information stores. The later project is sometimes referred as the "Name Space Browser," and sometimes as the "Open Dialog." To my knowledge, this project had nothing to do with the availability of the shell namespace extensions of the Windows Explorer. The equivalence of the Novell/WordPerfect "Open Dialog" (or "FileOpen Dialog") and "Name Space Browser" efforts are clear from contemporaneous Novell documents, e.g., "Name Space Browser (aka FileOpen Dialog)" (NOV-B01491965; See also NL2 0000071; Harral Exhibit 1).

6.C. Allegations Regarding MAPI

111. In his the discussion of MAPI in his report (Alepin Report at 105-143), Mr. Alepin repeatedly asserts that Microsoft did not disclose information regarding its modifications to its Messaging Application Programming Interface (MAPI) protocol, and consequently, non-Microsoft middleware developers could not interoperate with Microsoft's version of MAPI. For example, Mr. Alepin states "*The term 'MAPI Extensions' refers to Microsoft's undisclosed changes to MAPI that prevented interoperability among third-party products*" (Alepin Report at 115).

6.C.1. What is MAPI?

112. As I describe below, the acronym "MAPI" refers to several related concepts. It refers to two sets of messaging library functions (one is a subset of the other) that are used to create mail-enabled applications. It also refers to an RPC-based protocol (sometimes referred to as MAPI/RPC) designed by Microsoft and used for communication between the Microsoft Outlook client and the Microsoft Exchange server.[3] As Microsoft client and server products have evolved over time, so too have the MAPI APIs and protocols evolved. There are now at least four MAPI-related API sets/protocols used by Microsoft and others in what are sometimes called personal information management (PIM) systems. These include:

- **Microsoft Extended Messaging Application Programming Interface (MAPI 1.0, or Extended MAPI)** – "Extended MAPI" is an extensive set of functions that

[3] This protocol is now called the "Outlook-Exchange Transport Protocol."

developers can use to create email-enabled applications. This full function library is known as either Extended MAPI or MAPI 1.0 (and, unfortunately, sometimes as just "MAPI"). Extended MAPI allows a server-based messaging system to exercise control over the messaging system on the client computer, including creation and management of messages, and management of the client mailbox, service providers, etc.

- **Simple Messaging Application Programming Interface (Simple MAPI)** – Simple MAPI is a subset of 12 functions in MAPI that enable developers to add basic messaging functionality to Windows-based applications. Simple MAPI includes functions to support sending and receiving messages, including the ability to log onto the messaging system, compose new messages, add and resolve recipients, send messages, and retrieve and read messages from the inbox.
- **Collaboration Data Objects (CDO)** – Collaboration Data Objects ("CDO") was originally called "OLE Messaging" and later "Active Messaging". CDO is a Component Object Model (COM) wrapper for the MAPI library. CDO implements most, but not all, MAPI 1.0 functionality, although far more functionality than is included in Simple MAPI. Activities that can be accomplished using CDO include:
    - logging onto the messaging system with specific profiles or with anonymous authentication;
    - composing messages, addressing and resolving recipients, sending, receiving, and reading messages, adding attachments, automating replies,

- managing calendars, creating meetings and appointments;
- managing folders and messages within the information store; and
- managing Addresses, especially within the Personal Address Book ("PAB").

There are two distinct "flavors" of CDO, the MAPI based CDO.DLL and the Simple Mail Transfer Protocol (SMTP) based CDONTS.dll.

- **Web-based Distributed Authoring and Versioning (WebDAV)** – WebDAV is a set of extensions to the HTTP protocol that allows users to collaboratively edit and manage files on remote web servers based on international standards (IETF Request for Comments (RFCs) 2518 and 3253). Using WebDAV protocol methods, software developers can create, copy, delete, move, or search for resources in the Microsoft Exchange store, as well as set and search for resource properties.

113. With the exception of the Outlook-Exchange Transport Protocol, all of these APIs and protocols have been fully described (including examples of their use) in the Software Development Kits ("SDK") available for some time on the Microsoft Software Developer Network ("MSDN") web site, as well as through a subscription service. MAPI 1.0 also had an SDK (Software Development Kit) (MS 5041454). I regularly consult MSDN and find it a very comprehensive and useful source of information about Microsoft products, including various Windows operating systems.

114. The Outlook-Exchange Transport Protocol, although recently fully documented, has been historically used by Microsoft to capture some of the "business logic" of its PIM products. In my view, Microsoft has been justifiably reluctant to publish the business logic (for example, checking to see if a new appointment conflicts with an existing appointment) associated with Microsoft Outlook, because such innovations are what enable Microsoft to distinguish its products from those of competitors. In addition, any vendor was free to extend the MAPI protocols to support proprietary interfaces that would allow a messaging server to expose special behavior to its clients (MS-PCA-IA-EC 111853).

115. The MAPI API architecture consists of three parts: (1) client application program interfaces (APIs), which provide the client-to-MAPI link, (2) service-provider interfaces (SPIs), which provide the MAPI-to-messaging system link, and (3) the MAPI messaging subsystem within the Windows operating system that connects APIs to SPIs. None of these components represent "middleware," as Mr. Alepin uses that term, because they cannot conceivably be viewed as an alternative platform for developing general purpose applications.

116. MAPI thus provides a set of common function calls that, together with its messaging subsystem, act as brokers between the front-end (client) applications and the back-end (network-messaging) system. This dual interface helps ensure openness on both the client and server sides. Using MAPI calls, any messaging application can use any messaging service. Application developers are thus freed from messaging-system concerns, and messaging-system vendors are freed from application-specific concerns (MS7058541-61).

117. Simple MAPI and Extended MAPI provide the required API calls for messaging applications. These calls work with a second level of MAPI features built into the Windows operating system: the messaging subsystem. The MAPI messaging subsystem and a MAPI dynamic-link library (DLL) are responsible for dividing the tasks of handling multiple transport services providers (MS 7058541-61). Drivers for each transport service provider exist in the form of a Windows DLL to provide the interface between the MAPI messaging subsystem message spooler and the underlying backend messaging system(s) or services (MS 7058541-61).

118. The message spooler is similar to a print spooler except that it assists with the routing of messages instead of print jobs. The message spooler, MAPI.DLL, and the transport service providers work together to handle the sending and receiving of messages (MS 7058541-61).

6.C.2. MAPI-Related Allegations

119. Mr. Alepin's treatment of this subject is replete with statements for which no meaningful basis, foundation or citation is offered. I am unable to respond fully until such time as Mr. Alepin provides such basis, foundation or citation. In addition, Mr. Alepin in my view mischaracterizes the evidentiary record related to this issue, as I discuss below.

120. In his report, Mr. Alepin repeatedly refers to various components and instantiations of MAPI as "middleware" (Alepin Report at 105-106, 111, 113-114, 116-117, 124-127, 134-139, 141-143). Emblematic of this view is Figure 8 (Alepin Report at 113), in which Mr. Alepin appears to represent the MAPI32.dll as "middleware." MAPI and its components are not

middleware, as that term is commonly understood, or as it was defined in the Washington DC case.

6.C.3. Microsoft Outlook/Exchange Protocols

121. Mr. Alepin asserts "*These undisclosed extensions, when implemented in Microsoft's Exchange client, allowed Microsoft to achieve better functionality among its own products, and simultaneously prevented ISVs from achieving the same level of interoperability with Microsoft's products*." … "*For example, Novell's MAPI-compliant Groupwise server could not work with Microsoft's Exchange client or Outlook 97 desktop client in the same way as Microsoft's Exchange server could*" (Alepin Report at 115-116). Thus, Mr. Alepin's argument appears to be that Microsoft Exchange uses proprietary protocols to interoperate with Microsoft Outlook, e.g., the calendaring function, and consequently, non-Microsoft email products cannot fully employ this calendaring function. This is neither surprising nor uncommon. For example, GroupWise clients interact with GroupWise Servers in proprietary ways (Hume 25:20-26:1), and, to the best of my knowledge, Lotus Notes and Lotus Domino do as well.

122. In addition, vendors have been able to make their messaging server software interact fully with Microsoft Outlook, e.g., Lotus iNotes Access for Microsoft Outlook. As stated in the *iNotes Access for Microsoft Outlook Specification Sheet*, "The Microsoft Outlook user experience is unchanged with iNotes Access for Outlook; users simply work with their mail, calendar and task data on Domino instead of Microsoft Exchange. Familiar Microsoft Outlook

features are supported, including rich text, folders, and integration with Microsoft Office applications."[4] Plainly, Lotus was able to do what Mr. Alepin has suggested is not possible.

123. Finally, I also do not understand the relevance of these allegations, since Microsoft Outlook was not commercially released until after Novell sold its office productivity applications to Corel.

6.C.4. Allegations Regarding Groupware

Mr. Alepin's allegations with respect to MAPI are part of a general allegation that Microsoft's actions were intended to harm groupware competitors such as Novell GroupWise and Lotus Notes (Alepin Report at 105-106). "Groupware" refers to applications that allow collaboration among users such as email, calendaring, library functions, bulletin board functions, etc. In his report, Mr. Alepin alleges four general ways in which Microsoft caused harm groupware vendors. I discuss each of these in turn (although I am informed by counsel that there is serious doubt as to whether GroupWare is any part of this case given the lack of any mention of it in Novell's complaint or previous court decisions in the case).

124. **Bundling** – Microsoft released Windows for Workgroups 3.11 with Microsoft Mail and Microsoft Schedule+ included. The inclusion of Microsoft Mail and Microsoft Schedule+ with Windows for Workgroups 3.11 provided obvious benefits to users – they were able to use these personal productivity tools to their advantage. Microsoft also included basic email support in the initial release of Windows 95. This allowed users to easily email a document from inside an

---

[4] *iNotes Access for Microsoft Outlook Specification Sheet* (available at http://www.lotus.com/products/inotes.nsf/allpublic/00D522F4AA9933FF8525685E00572A9D).

application using a consistent interface. In 1995, email was a relatively new application for many users; having a basic email client included in Windows 95 gave novice users a simple mechanism to share documents with others (or themselves at a different location), without having to purchase (or select) another piece of software.

125. **Standards Participation** (Mr. Alepin's Allegations Regarding MAPI and VIM) – First, it is in my view perfectly natural for messaging interfaces designed and developed by Microsoft programmers to be tailored to the needs of Microsoft's messaging software and fully understood by Microsoft's messaging developers. I would expect a similar statement to be true about any software component developed by any group of software developers employed by any company. I see no reason to expect otherwise. Second, Mr. Alepin does not suggest in his report that MAPI is in any way inferior to VIM, and I am not aware of any basis for such a suggestion. In fact, as I describe below, VIM was largely "vaporware," a specification that even its proponents had not implemented in a software product.

126. In his report, Mr. Alepin asserts that "*In the fall of 1991, a group led by Lotus, WordPerfect, Novell and Apple approached Microsoft, inviting it to participate in what would become the Vendor Independent Messaging, or "VIM" group*." … "*Bill Gates declined the invitation to join VIM, either in its capacity as a vendor of messaging server-based software or in its capacity as vendor of applications that could be mail enabled*." (Alepin Report at 118). In fact, this "invitation" came in the form of a telephone call on the Friday afternoon before the Monday announcement of VIM (MS 5065146). According to Tom Evslin, who was responsible for MAPI within Microsoft, and who later became the head of Microsoft's Workgroup Applications

Division, the first Microsoft knew anything of substance about VIM was when Lotus made a public announcement about the VIM specification. Mr. Evslin testified that at the time VIM was "announced", Microsoft had already developed MAPI, and was about to ship:

Q. … And direct your attention down to the last paragraph on the page. It's talking about the hot topic being MAPI versus VIM.

A. Uh-hum.

Q. What do you remember about that?

A. This is before MAPI had become an industry standard. MAPI –

Q. Okay.

A. -- at the time was a Microsoft set of APIs, and a group of Microsoft competitors which included Lotus, Apple, Novell, I think IBM had joined together and proposed a competing standard which was called VIM or a competing set of APIs rather called VIM. There was -- at first there -- so both were -- nobody -- when I say propose them, I really mean announce them. Nobody had proposed that these become official standards. It's just that this group, the VIM group, had said we're going to support this. Microsoft was already supporting MAPI. At the time of this memo, that was the state of the two. VIM was a vapor standard. It didn't exist. It wasn't implemented in any products. MAPI was implemented in the version of MS Mail that I believe was about to ship at this time. I'm not quite sure on the chronology.

Q. When you say "VIM was a vapor standard," what does that mean?

A. There were no products that embodied it. So it was -- I -- actually it was not a standard at all, but it was a vapor set of APIs in that suppose you said okay, I like VIM, I'm going to develop to VIM, there wasn't anything -- there were no products that you -- would support you.

(Evslin Dep. at 114-115)

A. Lotus suddenly was an adherent of this API despite not having implemented it in any of their products. So they were the ones who had an opportunity to. In general, if you want people to support an

> **API, if you're serious about an API, you tell all major developers about it. We had done that with MAPI, for example. Lotus had never approached us before they had the press conference announcing VIM. So it was hard for us, again, to take it seriously that they were doing anything except trying to stop any momentum for MAPI, and then the technical -- VIM was -- was technically deficient, and I'm sure, although I don't recollect it directly, that we explained those technical deficiencies to people who came by the press suite.**

(Evslin Dep. at 117-118)

127. Mr. Alepin states that "*Moreover, after manufacturing a precedent for bundling MAPI with the operating system, and committing to make MAPI cross-platform, Microsoft announced that it would not provide MAPI for the Macintosh, contrary to its January 1994 agreement with Apple*" (Alepin Report at 126). Mr. Alepin offers no primary source document that supports the existence of such an agreement (his only "source" is an unattributed quotation in a March 23, 1995 letter from Michael D. Zisman of Lotus to Jim Manzi (IBM 7510251964-70)). In addition, contemporaneous news coverage of the agreement that was reached between Microsoft and Apple identifies no agreement by Microsoft to "provide MAPI for the Macintosh". The January 7, 1994 *PR Newswire* states that "The agreement also includes Apple's plan to provide a MAPI service provider that will allow applications and users of Windows and MAPI to access and use Apple's PowerShare Collaboration Servers." The statement goes on to describe Microsoft's involvement: "Under the agreement, Microsoft will provide a personal gateway that will allow AOCE-based PowerTalk users and applications on Macintosh and PowerBook computers to access and use Microsoft's messaging and directory services provided by Microsoft's future information management products." (*PR Newswire*, January 7, 1994; MS 5056342). It appears that Apple subsequently "farmed out" its responsibilities under the agreement to Transend

Corporation. (*PR Newswire*, May 8, 1995). Less than two weeks later, a representative of Microsoft stated that Microsoft remained "committed to a multi-platform messaging API through its support for Common Mail Calls." (*Electronic Messaging News*, Vol. 7, Issue: 10, May 17, 1995.) Common Mail Calls were a functional subset of MAPI also known as "Simple MAPI".

128. **Tying Messaging to the Operating System and its Applications and Interface Disclosure** – Mr. Alepin alleges in his report that that

> "*Microsoft sought to eliminate the threat posed by this new category of software through a series of tactics that are strikingly similar to those it employed against Netscape and DR-DOS, including:*
> - *committing that its MAPI middleware would be open;*
> - *committing that its MAPI middleware would be cross-platform;*
> - *bundling its MAPI middleware in its Windows operating systems;*
> - *announcing that Chicago, a future operating system, would ship with MAPI; and*
> - *exercising its control over the desktop to place its messaging icons on the desktop*."
>
> (Alepin Report at 105-106).

129. Putting aside Mr. Alepin's unsubstantiated claims of similarity to unrelated subject matter, I have already discussed the way in which the APIs referred to as MAPI were in fact an open specification (albeit one that was designed to be extensible), and the user benefits of including basic messaging within the operating system. Since MAPI offered functionality that could be beneficially used by many different applications, it is reasonable for Microsoft to have placed this functionality in the operating system for the benefit of all application developers, and to have "evangelized" that functionality to ISVs. Also, as I have already discussed, including more functionality in operating systems was part of their natural evolution, as demonstrated by the continuous addition of features to non-Microsoft operating systems such as Apple's Macintosh

operating system and IBM's OS/2 in the relevant time period. Further, including functionality within Windows did not affect the operation of products offered by Lotus, Novell, or any other groupware vendor.

130. In contrast to Mr. Alepin's views on this issue, it appears to me that all of the APIs necessary to develop a MAPI-based messaging client were published by Microsoft. Microsoft may have extended the functionality of MAPI for its own purposes (as Novell could have done as well), but that did not in any way limit the ability of ISVs to use the published MAPI interface. As Mr. Evslin testified:

> Q. Just to be clear, the MAPI was published.
>
> **A. That's correct.**
>
> Q. And MAPI was what again now?
>
> **A. MAPI was the messaging APIs, and so all of the APIs necessary to a messaging client were published. We had made a prior commitment to that, as far as I know, that commitment was kept. There was never any question about whether we were going to keep that commitment once it had been made. Those were all the APIs necessary to write a messaging client. The APIs that were under discussion were APIs that you wouldn't need to write a messaging client in order to do the messaging functions, send mail, read mail, find mail, but that provided a tighter user interface integration between Windows and the mail client, but you could write a fully featured mail client with just the MAPI APIs. That was the purpose of the MAPI APIs.**
>
> (Evslin Dep. at 260-261)

131. In fact, Lotus also considered adding functionality to MAPI for its own purposes. As Barry Briggs of Lotus wrote in an internal Lotus memorandum: "…*This in turn signifies MS is going full steam ahead against cc:Mail, and that our strategy of being a MAPI client "plus" is*

precisely their strategy as well. A very interesting possibility for us to exploit this situation is to publicly extend MAPI ourselves". (IBM 7510366194).

132. Therefore, in summary, Mr. Alepin's allegations with regard to MAPI in particular, and to groupware in general, are inconsistent with the record that I have reviewed.

6.D. Allegations Regarding Windows 95 Logo Certification

133. In Mr. Alepin's discussion of allegations regarding the Windows 95 Logo Certification Program (Alepin Report at 16, 144-155), he suggests that Novell (and other ISVs) were unreasonably burdened by a requirement to test their applications on Windows NT in order to obtain the right to use the Windows 95 Logo on the packaging of their applications (Alepin Report at 16). Contrary to what Mr. Alepin's says, there were substantial benefits to both ISVs and consumers for Microsoft to impose this requirement, as I describe below.

134. The summary of the Windows 95 Logo requirements in Mr. Alepin's report (Alepin Report at 147), while accurate in some respects, is overly simplistic. Below I quote a summary of the actual requirements from the MSDN web site.

"Summary of the Logo Requirements

There are five basic requirements your application must meet to qualify for the Windows 95 logo, no matter what type of application it is:

- Win32 executable

  Your application must be a Win32 executable file using the PE (Portable Executable) format. This requirement is satisfied automatically, because Visual C++ for the Intel platform always produces executable files in the PE format.

  If you have a 16-bit application that you need to port, see Porting 16-Bit Code to 32-Bit Windows.

- UPI and Shell support

  Your application must follow the Windows 95 application setup guidelines, register large and small icons, and use system color and metrics. It's also recommended that your application provide context menus through the right mouse button, use the common dialogs and controls, and follow the user interface guidelines set forth in the *Windows Interface Guidelines for Software Design* on the MSDN Library CD.

  The section Following UPI Recommendations below discusses how to add this support using MFC.

- Windows NT compatibility

  Your application must run successfully on both Windows 95 and Windows NT 3.5 (or greater). If your application uses functionality specific to either operating system, its behavior must degrade gracefully when run on the other operating system (that is, an execution error should not result).

  Meeting this requirement depends on your use of API functions specific to Windows 95 or Windows NT. For a list of these APIs, see the paper "Diving into the Requirements for the Windows 95 Logo" on the MSDN Library CD.

- Long file name support

  Your application must be able to accept and store long file names and display them in its title bar, in dialogs and controls, and so on.

  The MFC library supports the use of long file names, so you can pass a long file name to any MFC function taking a file name as a parameter. You should also use the MFC common dialog class **CFileDialog** when requesting a file name from the user. Make sure that any file name-handling code of your own can handle file names longer than eight characters and file names containing spaces or other special characters. For more information on long file names, see Supporting Long File names on the MSDN Library CD.

- Plug and Play support

  This is recommended but not required. Your application should respond appropriately to the **WM_DEVICECHANGE**, **WM_DISPLAYCHANGE**, and **WM_POWERBROADCAST** messages, which signal changes in peripheral devices, the display resolution, or the system power status, respectively.

  The MFC library responds to the **WM_DISPLAYCHANGE** message and resizes windows and toolbars according to the new system metrics. How an application should respond to the **WM_DEVICECHANGE** and **WM_POWERBROADCAST** messages depends on the specific application; for an example of the most common response to these messages, see the Plug-and-Play awareness component in the Gallery.

There are also three other requirements your application must meet if it is file-based (that is, if your application's primary purpose is to create, edit, and save files):

- UNC path support

  Your application must support Universal Naming Convention (UNC) paths. That means your application must be able to accept and store paths of the form "\\server\share\directory" directly, without requiring the user to assign a drive letter to the server beforehand.

The MFC library supports the use of UNC paths, so you can pass a UNC path to any MFC function taking a path as a parameter. You should also use the MFC common dialog class **CFileDialog** when requesting a file name from the user. Make sure that any pathname-handling code that you write can handle double backslashes instead of a drive letter at the beginning of a path.

- OLE support

  Your application must either be an OLE container or an OLE server, or both. In addition, if it's an OLE container, it must act as a target for drag-and-drop operations, and if it's an OLE server, it must act as a source for drag-and-drop operations. It's also recommended (thought not required) that you support Automation and provide summary information with your documents.

  The section Adding OLE Support discusses how to add this functionality, using MFC.

- MAPI support

  Your application must include a Send Mail command on the File menu to enable the user to send the current document as a piece of mail, using MAPI or the Common Messaging Call (CMC) API.

  The section Adding MAPI Support discusses how to add this functionality, using MFC.

There are certain additions and exceptions to the previous requirements, depending on the type of application you're developing; these additions and exceptions are described in the "Logo Criteria" document.
The remainder of this section on Windows 95 logo compliance describes the changes made to the sample application to make it meet the Windows 95 logo requirements, as well as certain user-interface recommendations; you can use this discussion as a guide for modifying your own MFC application."
(http://msdn.microsoft.com/en-us/library/aa296573(VS.60).aspx)

135. Each of these Windows 95 Logo certification requirements was beneficial to ISVs, to consumers, or both, as I enumerate below.

Win32 Executable

136. As I described above, some existing applications took advantage of MS-DOS and Windows 3.x behavior in ways that were unsafe. Since Windows 95 continued to support a real mode DOS component, and was designed to be backward compatible with the vast majority of Windows 3.x applications, requiring that developers use the Win32 APIs helped ensure reliable applications that would continue to function as the operating system was upgraded. This was a benefit to both ISVs and consumers.

UPI and Shell Support

137. This requirement ensured that applications presented a user interface and installation process that would be familiar and consistent for users across applications. The installation process was a common source of problems for inexperienced users, generating large numbers of support calls, so making the process familiar to users was good for both users and for ISVs. Adopting the additional recommended practices related to this subject strengthened the consistency among Windows 95 applications.

Windows NT Compatibility

138. Window 95 included a real mode DOS component in order to maintain backward compatibility with the large installed base of Windows 3.x and MS-DOS applications and device drivers. This backward compatibility exposed Windows 95 to the many "hacks" that developers of Windows 3.x and MS-DOS applications and device drivers had previously employed to extract the best possible performance from the hardware and operating system. Real mode applications routinely manipulated directly critical operating systems structures to avoid the overhead of calling the operating systems APIs designed to perform those manipulations safely. As microprocessor speeds improved, such machinations were no longer necessary, and their risks began to substantially outweigh their benefits. Windows 95 was developed for machines of sufficient power and capacity that applications could be designed to run safely on the operating system. This was a significant benefit to both consumers and ISVs. Consumers had applications that did not crash unexpectedly when they upgraded their operating system, and ISVs had applications that were easier to develop, test, and support. However, these goals would not be

realized if applications continued to manipulate operating system structures, or directly access peripheral devices, even if such actions only occurred in the Windows 95 real mode component.

139. In contrast with Windows 95, Windows NT was developed from the ground up as a new operating system, and did not preserve any vestiges of MS-DOS in a real mode DOS component. In designing Windows NT, Microsoft made a clear decision to forgo backward compatibility with applications that relied upon the unprotected architecture of MS-DOS and Windows 3.x. Thus, an application running on Windows NT was unable to manipulate operating system structures, or directly access peripheral devices. As Mr. Alepin acknowledges: "*At their core, the kernel, Windows NT and Windows 95 were very different. Windows 95 evolved from earlier versions of MS-DOS and Windows. Over time, developers learned special ways to use the Windows kernel, its memory allocation, and other components, certain of which might not have been the interfaces Microsoft originally intended. These techniques were "baked into" the ISVs' applications. On the other hand, Windows NT started from scratch, targeting different customers who demanded more security. More importantly, Windows NT was designed to be independent of any one hardware platform*" (Alepin Report at 150). It was these "special ways to use the Windows kernel, its memory allocation, and other components" that were threats to the stability of the operating system. In order to help ensure that Windows 95 would remain stable, and that applications would continue to run when the operating system was upgraded (a clear benefit to all concerned), a mechanism was required that would test for applications that exhibited such this dangerous behavior. Since Windows NT also supported the Win32 API set exported by Windows 95, but did not allow applications to access the operating system in unsafe ways, Windows NT represented a convenient mechanism for identifying "well-behaved" applications.

140. Requiring Windows 95 applications to be tested on Windows NT as part of the Windows 95 Logo Certification Program was a benefit to ISVs, who had a clear, convenient, and readily available mechanism to evaluate both their legacy and new applications, and to consumers, who could be assured that applications bearing the Windows 95 logo were less likely to crash their personal computer, and to continue to work when they upgraded their operating system. There was an added benefit for those users who ran Windows 95 at home and Windows NT at work: they were able to run common applications in both environments. Finally, although the user base of Windows NT was relatively small at the time, it was growing. Thus, the ability of an application to run on both Windows 95 and Windows NT expanded the pool of potential purchasers of the application.

Long File Name Support

141. This was a clear benefit to consumers, and a distinguishing feature for ISVs who wanted to support modern file system interaction. The so-called eight-dot-three file names possible in MS-DOS, i.e., washingt.doc, were difficult and limiting (and probably annoying) for users. The ability to have file names of up to 255 characters, including spaces and numbers, was a significant improvement.

Plug and Play Support

142. Although this was only a recommendation, rather than a requirement, applications for Windows 95 that responded to changes in peripheral devices, the display resolution, or the system power status provided a considerable benefit to consumers. For example, the ability to detect when a portable projector is being used, and resizing the application's user window

accordingly, saves users from having to manually configure their personal computer when making presentations.

For File-Based Applications Only:

UNC path support

143. UNC (Universal/Uniform Naming Convention) paths represent a common way to describe the location of a network resource, such as a shared file, directory, or printer. The UNC path syntax for Windows systems has the generic form: \\ComputerName\SharedFolder\Resource, in contrast to the need to specify the drive letter of a local file system, such as C:\File.

144. This requirement ensured that the user would be able to interact with local and remote files in a consistent manner, a clear benefit.

OLE Support

145. Object Linking and Embedding (OLE) refers to technology for embedding objects, such as pictures or spreadsheets, into a document, such as a Word document. OLE allows the embedded object to be manipulated, e.g., an embedded spreadsheet still works like a spreadsheet. This requirement ensured that Windows 95 users would be able to "drag and drop" and "cut and paste" across applications, a clear benefit to consumers and a distinguishing feature for ISVs.

Simple MAPI Support

146. This requirement ensured that Windows 95 applications could easily send a document via email using a consistent interface. The alternative would have been to require users to open a separate email application and attach the document to the email, a much more complicated and

cumbersome process. In 1995, email was a relatively new application for many users; the requirement that applications support Simple MAPI gave those novice users a simple mechanism to share their documents with others (or themselves at a different location) via email.

147. Mr. Alepin also writes that "*Among these requirements was ISV support for strategic technologies that the industry had not otherwise adopted as standards. For instance, the logo required applications to support MAPI or CMC, rather than VIM; OLE 2.0, rather than OpenDoc; and Windows NT, rather than Novell's NetWare. Applications using these strategic technologies would be more tightly locked to the Windows 95 platform than if they used the rival technologies, which were often platform independent*" (Alepin Report at 147-148). I have already addressed issues related to MAPI and VIM, and the substantial benefits of the Windows NT testing requirement. I also note that Windows 95 did support Novell NetWare; Windows 95 came with a NetWare client developed by Microsoft, and it was easy to obtain Novell's NetWare client and install it on Windows 95.

148. Mr. Alepin also appears to imply that Microsoft gave preference to OLE over "OpenDoc[5]" in the Windows 95 Logo Certification Program. Since OLE functionality had been shipping for years prior to the development of the OpenDoc specification, it is not surprising to me that Microsoft was not enthusiastic about revamping their office productivity applications to support OpenDoc.

---

[5] Unfortunately, the term "OpenDoc" refers to two different things. The first use of the term is as described here. More recently, the term as been used to refer to a document format. In this context, I believe that Mr. Alepin is referring to the first usage.

149. OLE is a set of standard COM (Component Object Model, a Microsoft-created standard introduced in 1993) interfaces that enable users to create compound documents by linking and embedding objects (components) into container applications. Mr. Alepin implies in his report that it was somehow inappropriate for Microsoft to incorporate OLE technology into the operating system. (Alepin Report at 147-148). However, the incorporation of OLE into Windows benefited both developers (for whom application programs were easier to write) and consumers (for whom application programs were easier to use; for example, it was easier to "cut and paste" between documents, since documents created by different applications have a common "cut and paste" interface). Scot Alan Kliger, a software developer who had worked at IBM, WordPerfect, and Lotus acknowledged that it was not only appropriate, but beneficial, for Microsoft to move OLE functionality into the operating system:

> **A. As an application developer, it's not unusual at all for software that is developed in one application that can be made into a general purpose way that is beneficial to either other applications -- so, you know -- and I'm sure we'll talk about my experience at Lotus at some point, but at Lotus, it was common to take a component and make it what was called "common code" that would then be shared between the applications. Now, since Lotus didn't have an operating system, it really wouldn't make much sense to put it in the operating system, but for Microsoft, it made all the sense in the world to put it in the operating system, and by doing that, make it available to all of the application developers to give them the benefits of the rich functionality that had been developed, as well as an easy way for them to implement the functionality so that it was consistent across any application. And I think that's really an important point because, as an application developer, you know, for something like Object Linking and Embedding, for WordPerfect, we had no idea, if a word processing document was going to get imbedded in a spreadsheet, whether it was going to be Borland's spreadsheet or Microsoft's spreadsheet or Lotus's spreadsheet. And had we had to develop functionality that was going to work with each of them and then have to go, you know, cut a deal with each company in order to say, Okay, well, when we're working with you, it's going to work this way, it would have been impossible to navigate. But by having one**

> **standard with which we could know that as long as we implemented it that way, it was going to work, then that made our life a lot easier.**
> (Kliger Dep. at 125-127)

150. Further, OLE was not isolated functionality; it was a small part of the large suite of functionality provided by the Windows operating system through its published APIs. Software developers creating applications for Windows were likely already to be making extensive use of these APIs, so I would not expect the use of OLE functionality to represent a significant stretch for these developers.

6.D.1. Novell's Request for Exemption

151. Microsoft provided the means for ISVs to request exemptions to the Windows NT Logo certification requirement if "the application relies on functionality that is significantly different in architecture between the two platforms and that functionality cannot easily degrade gracefully" (MSC 00700613). On March 6, 1995, Novell requested such an exemption in an email from Mark Calkins of Novell to Brad Silverberg and Brad Chase of Microsoft (NOV 00019380-82). Mr. Chase responded by email on April 3, 1995, stating that "At this point in time, we do not believe the issues you raise constitute significant enough architectural issues between the Windows NT and Windows 95 to warrant an exception being granted" (MSC 00700615), enumerating the reasons for this decision point by point, and suggesting specific ways to address the issues that Novell had identified. I find it noteworthy that WordPerfect was in some cases apparently exhibiting precisely the kind of unsafe application behavior that the Windows NT requirement was intended to detect. As Mr. Chase stated in his reply:

> "Your development teams should not exploit some observed functionality as it appears they are doing from several of your questions. Exploiting such features will most likely cause these applications to break in future releases of the operating system as we can only commit to supporting documented behavior as we move forward" (MSC 00700613).

152. Mr. Alepin's states that he "*analyzed both Novell's original analysis and Microsoft's reply and, in my technical opinion, have concluded that Novell's assessment of the reasons for its inability to comply with the logo requirement was correct. Architectural differences between Windows 95 and Windows NT - coupled with ambiguities as to whether the application needed to be compatible with Windows NT 3.5 or 3.5 1 - were the root cause of Novell's issues*" (Alepin at 153). Although Mr. Alepin rejects Microsoft's response in its entirety, it appears that someone at Novell agreed with at least some of the points made by Mr. Chase (NOV-B01152591 – 600).[6]

153. I also note that Mr. Chase, in his response to Novell, did not foreclose the issue. At several points in his letter, he makes clear that the decision not to grant an exception to the requirements of the Windows 95 Logo Certification Program remained open for discussion. For example, the single example of his "analysis" that Mr. Alepin offers in his report omits the last sentence in Mr. Chase's response to the specific point about the Registry, which reads, in its entirety:

> "RegSaveKey and the related APIs are not designed to be binary compatible between Windows 95 and Windows NT. It is true that the data employed by these APIs is not the same between the platforms but it wasn't meant for this. The data is meant to be black box and not portable. This should not cause problems for mainstream applications. ***We can discuss this issue in more detail in a follow-up conference call, if you would like***." (MSC 00700614, emphasis added)

[6] This document is an undated, unsigned, and, to my knowledge, unsent response to Mr. Chase's letter.

154. Other portions of Mr. Chase's letter also makes it clear that the decision not to grant an exception to the requirements of the Windows 95 Logo Certification Program remained open for discussion:

> "If in our discussions we decide that this exception was to be granted to Novell, you would need to make the difficult business decision to release a product to market that did not support the same platforms that many of the leading applications will."
> …
> "In closing, Microsoft wants to be flexible with the logo if a product cannot run on Windows NT because of significant architectural differences between the Windows NT and Windows 95. If an application runs into significant architectural differences then our policy is not to withhold the Logo for that application. If the developer codes something incorrectly however, then the problem needs to be corrected before the Logo is issued. This is simply a question of meeting customer expectations with respect to support between the two operating systems."
> …
> "I would be glad to have a conference call between our teams should you have any additional questions. Brad Struss will be glad to set this up for you if you wish.
>
> Mark, we appreciate your continued support of Windows and in particular Windows 95. I believe it is the best thing for your customers to build applications that will run on both Windows 95 and Windows NT. Of course this is your call and as you know, participation in the Windows 95 Logo Program is optional and by no means required to ship a great Windows 95 application.
>
> Please let us know if you would like to speak further and if there is anything else we can do to be of help." (MSC 00700613 - MSC 00700618, excerpted)

155. I am not aware of any Novell response to Mr. Chase that was ever sent, nor does the record reflect any effort by Novell to engage in the dialog suggested by Mr. Chase.

156. Mr. Alepin summarizes his opinion by stating "*In sum, in my technical opinion, it is impossible to view Novell's problems as meritless. Accordingly, it is not possible to construe Microsoft's response to Novell's issues to be based on an honest difference of technical opinion*" (Alepin Report at 154). Leaving aside the flawed logic of this statement, Mr. Alepin offers no

basis whatsoever for the view that "it is not possible to construe Microsoft's response to Novell's issues to be based on an honest difference of technical opinion." Moreover, I have no difficulty reaching the conclusion that there was indeed such a difference of technical opinion. Indeed, for the reasons enumerated above, Microsoft had the stronger side of the argument.

157. Mr. Alepin also writes: "*Despite waiving NT compatibility for 10% of the products that sought Windows 95 logo certification, Microsoft refused to grant such a waiver to Novell*" (Alepin Report at 154). In addition to mischaracterizing Microsoft's response, Mr. Alepin's sole reference for this assertion is an industry advocacy document entitled "Competition in the Network Market: The Microsoft Challenge" written by "by the staff of the SPA [Software Publishers Association], under the direction of Lauren Hall, chief technologist, with input from many SPA member companies" (NOV00686920-959). I have no way of knowing whether the statements contained in that document are correct, especially because the portions of the document upon which Mr. Alepin relies do not provide any support for the assertions made.

158. Finally, I note that the Novell **YES** logo certification program required ISVs seeking to certify an application as "NetWare compatible" were required to also be compatible with UnixWare and Lananlyzer (NOV-25-000119).

159. In summary, the Windows 95 Logo Program offered direct benefits to both users and ISVs. Further, Microsoft's criteria for deciding which applications qualified for the right to use the Windows 95 logo are, for the reasons that I have enumerated above, technically sound. Finally, I have seen no evidence in this case that Novell (or any ISV) was singled out for unfair treatment in the administration of the Windows 95 Logo Certification Program.

6.E. Allegations Regarding Printing Interfaces in Windows 95

160. Mr. Alepin asserts that "*Microsoft impaired Novell's ability to provide "background printing" for its business productivity applications by committing to deliver and subsequently not providing the ability to utilize Windows 95 APIs to create a custom print processor*" (Alepin Report at 17). Mr. Alepin goes on to asserts that "*Microsoft continued to assure Novell that the functionality would be available until the last moment, when Microsoft informed Novell that the functionality would not be available in the initial release of Windows 95*" (Alepin Report at 17).

161. First, I note that in his discussion of this issue, Mr. Alepin repeatedly cites to an "Interview of Stuart Jensen" that apparently took place on April 24, 2009 (Alepin Report Footnotes 55, 749, 755, 776, 783, 784, 785). Moreover, many of the assertions in Mr. Alepin's report, for which this interview forms the sole basis, contain subjective opinions, and are not simply statements of fact. I doubt whether Mr. Jensen is equipped to provide information about Microsoft's motivations or intentions in 1994. As far as I am aware, Mr. Jensen has never testified under oath about any of the things he purportedly told Mr. Alepin. In addition, Mr. Alepin has provided no record of his "interview" of Mr. Jensen; as a result, I have no idea what questions were asked, and what answers were given. Absent such information, I am unable to respond fully to Mr. Alepin's opinions regarding this issue until such time as Mr. Alepin makes this information available.

162. In addition to offering no meaningful foundation for many of the statements in his report regarding the printing interfaces in Windows 95, Mr. Alepin repeatedly cites to documents that do not support the assertions made in his report, as I discuss below.

163. I have reviewed the evidentiary record as it relates Mr. Alepin's allegations to printing. I conclude that Mr. Alepin's allegations are not supported by the record and timeline of events. Mr. Alepin repeatedly states that Microsoft "promised" (Alepin Report at 8, 156, 159, 160, 162, 164) and "evangelized" (Alepin Report at 156, 157, 159, 164) certain printing-related functionality in Windows 95. The basis for these assertions apparently includes an undated[7] trip report from an unidentified Novell employee (NOV-B01786593-609), non-specific references from the March 1995 Windows 95 Device Driver Development Kit (DDK) (excerpts of this document appear to be found in NOV-B01645812-954), an undated document apparently discussing intellectual property matters related to the planned sale of the "Business Applications Division" (NOV00431599-624), and the *Microsoft Windows "Chicago" Beta 1 Reviewer's Guide* (MSC 00762731 – 998), which, although undated, presumably was available upon or near the release of Beta 1 of Chicago (6/10/94). This document states on its cover that:

> "The information discussed in this guide is based on features and functionality present either in the Beta-1 release of Chicago, or planned for a future release. ***The discussion of Chicago herein, does not represent a commitment on the part of Microsoft for providing or shipping the features and functionality discussed in the final retail product offerings of Chicago***" (MSC 00762731, emphasis added)

164. No one reading this document would be likely to miss this statement, which is entirely consistent with my experience with beta testing software throughout my career. All other documents cited by Mr. Alepin related to Microsoft's alleged "promises" are dated in 1995. These documents, and their relevance to the actual timeline of events, are discussed below.

---

[7] Based upon the document's contents, it was apparently written in January, 1993.

165. The second set of evidence relied upon by Mr. Alepin to support his assertions about Microsoft's supposed "promises" are documents that purport to record the substance of three conversations that took place between Don Miller at Microsoft and Stuart Jensen at Novell/WordPerfect in the period February 1995 to June 1995 (NOV 00516222-25). In particular, Mr. Alepin several times quotes from and cites to NOV 00516225, a chart prepared by an unknown person at Novell/WordPerfect representing things allegedly said by Don Miller at Microsoft on February 28, 1995, March 14, 1995, and June 29, 1995. Mr. Alepin quotes the following text as evidence of Microsoft's alleged commitment:

> "We can use SetJob(), via the pDataType parameter of the JOB_INFO_ 1 structure, to change the document data type to a custom data type. However, in the current version this will not work because the OS has the print processor hard coded. It should be fixed by the M8 release. (Don Miller)" (NOV 00516225).

166. This chart, Bates numbered NOV 00516225, is in fact a transcribed summary of another more detailed document (NOV-B01426539-40), entitled "Microsoft Questions," that is also cited by Mr. Alepin. Unfortunately, essential information was apparently lost in the creation of the summary chart quoted above. The original document states:

> **"Can we define our own data type to put in the pDataType field of DOC_INFO_1?**
> February 28, 1995 Don Miller -Yes, should work, but the Print sub-system currently has the print processor hard coded. This should change with the M8 build?"

167. A key difference between the two documents is that, in document NOV-BO1426540, a question is asked: "This should change with the M8 build?". This question implies uncertainty, and does not support the notion that Microsoft promised anything. In document NOV 00516225, an unequivocal statement is made: "It should be fixed by the M8 release." This statement appears to misrepresent what Mr. Miller actually said, if indeed he said any of the things attributed to him in the Novell documents.

168. In addition, the record reflects that, in the Fall of 1994, Novell/WordPerfect was in the very early design stage for the printing functionality to be in included in its new 32-bit office productivity applications for Windows 95. A nine-page rudimentary design document entitled "*WordPerfect Printing for 32-bit Windows*" (NOV-B01396671-79) indicates that it underwent four reviews from August 23, 1994 to September 1, 1994. The abstract of this document reads:

> "Product: WordPerfect Printing for 32-bit Windows
> Version: 6.1
> Machine/OS: 32-bit Windows (Chicago and NT) on both RISC and CISC platforms
> This design deals with modifications needed to be made to WordPerfect Printing to take advantage of new 32-bit features like enhanced metafiles, new print API, named pipes, and multithreading to improve performance and differentiate 32-bit printing for Wpwin6.1 from 16 bit printing." (NOV-B01396671)

This document also indicates the need for substantial product research yet to be done (NOV-B01396671), and that no software code had yet been written (NOV-B01396678-79).

169. The record also reflects that, at least as late as February 9, 1995, Novell/WordPerfect still had only an incomplete project development plan for the printing functionality to be included in its Windows 95 applications. The February 9, 1995 document entitled *PerfectFit Print Services*

Project Development Plan, Version 1.3[8] (NOV-B01433967-87) contains only the most rudimentary description of the printing functionality that Novell/WordPerfect was planning for its Windows 95 applications.

170. A third design document, *QCG (Qcode Generation Interface) Design Document, Version 1.6*, dated June 12, 1995[9] (NOV-B01426796-870), contains considerably more detail, but also reflects that there were many changes in the design of the Qcode generation interface (a component of the printing functionality planned for Novel/WordPerfect's Windows 95 applications) from February 15, 1995 through June 12, 1995.

171. Even taking the documents relied upon by Mr. Alepin at face value, I cannot see how Novell/WordPerfect could have been injured by anything Microsoft said about the printing capabilities of Windows 95. Given that Novell/WordPerfect was told, at least at the time of the Beta 1 release of Windows 95, that they should necessarily rely on any given functionality to be present in the final release of Windows 95; and was told, at least as early as February 28, 1995, that the original vision for Windows 95 print support might not be fully realized; and was told definitively on June 29, 1995 that Windows 95 would not contain all of the same printing functionality as Windows NT; it is difficult to see how Novell/WordPerfect was disadvantaged. That is especially true because the printing functionality to be included in their Windows 95 applications was still being developed during the February to June 1995 time frame.

---

[8] The cover of this document incorrectly displays a date of February 9, 1994, but both the Document Change History and internal page numbers clearly establish the document's date as February 9, 1995. Mr. Alepin apparently did not notice this error, and incorrectly states that this document was prepared in "early 1994" (Alepin Report at 159). Version 1.0 of this document was apparently released on January 5, 1995.

[9] The first draft of this document is dated February 15, 1995 (NOV-B01426797).

6.F. Miscellaneous Allegations in Alepin Report

172. In the final ten pages of his May 2009 report, Mr. Alepin makes a series of largely unsubstantiated assertions related to issues that have been raised in other cases involving Microsoft, including allegations related to Java, Netscape and Internet Explorer; disclosure of interface information; ActiveX; Windows help files; various "undocumented interfaces"; OLE and OCX; RTF; and what Mr. Alepin calls the "OS/2 Head Fake." Mr. Alepin's cursory treatment of these issues in his report appears to ignore or mischaracterize much of the substantial evidentiary record related to these matters. In prior cases, I have written hundreds of pages of analysis related to these and other allegations, and I incorporate them here by reference. To the extent that Mr. Alepin later adds substance to his barebones allegations – which strike me as probably irrelevant (or, at best, highly tangential) to the issues in this case – or those issues are raised at trial, I will rely upon that previous analysis in responding to the allegations rather than setting it forth again in full here.

John K. Bennett

June 24, 2009

Appendix A: Resume of John K. Bennett

Director of the ATLAS Institute
Archuleta Professor of Computer Science
Professor of Electrical and Computer Engineering
Professor in the Interdisciplinary Telecommunications Program
University of Colorado at Boulder
Boulder, CO 80309
(303) 492-4398
jkb@colorado.edu

Education

- **University of Washington**
  Ph.D. in Computer Science, March 1988 (Thesis Advisor: Edward Lazowska)
  M.S. in Computer Science, June 1983
- **Rice University**
  MEE, May 1974
  BSEE, May 1973

Professional Engineering Registration: State of Texas, P.E. No. 75525

Memberships
ACM, SIGOPS, SIGARCH, SIGMETRICS, SIGCOMM, IEEE, IEEE Computer Society.

Research Interests
Information and Communication Technology for Development; Distributed, pervasive, and mobile computing; parallel shared memory computing, including: distributed information management, high performance networks, parallel programming environments, DSM systems, and parallel operating system / architecture interaction; Digital inclusion; Engineering education and outreach.

Experience

University of Colorado	Director of the ATLAS Institute, 2007- Associate Dean of Engineering for Education, 2002-2007 Professor of Computer Science, 2000- Professor of Electrical and Computer Engineering, 2000- Professor in the Interdisciplinary Telecommunications Program, 2007-
Rice University	Master of Richardson College, 1996-2000 Associate Professor of Electrical and Computer Engineering, 1994-2000 Assistant Professor of Electrical and Computer Engineering, 1988-1994
University of Washington	Visiting Associate Professor of Computer Science, 1996 Predoctoral Lecturer in Computer Science, Winter, 1983 Research Assistant in Computer Science, 1980-1982, 1987 Research Assistant in High Energy Physics, 1979-1980
Pacific Mtn. Research, Inc.	President, 1982-1987

US Navy Sr. Ship Superintendent, Puget Sound Naval Shipyard, 1977-1979
Electrical Officer & DCA, USS Paul F. Foster (DD-964) 1974-1977

Research Funding

- AIR – Advancement through Interactive Radio. Microsoft Corporation Digital Inclusion Program ($150,000), 2006-2008.
- Automatic Distributed Unmanned Vehicles Coordination and Control. Raytheon Corporation ($25,000), 2005-2006.
- Rehabilitation Engineering Research Center On Recreational Technologies And Exercise Physiology: Project Rec-Tech. NSF ($4,500,000), 2002-2007
- Collaborative Research: Affinity Directed Mobility for Location Independent Data Access. NSF ($346,995), 2002-2005.
- PDA-Enhanced Speech Treatment for Parkinson Disease. NIH ($302,701), 2001-2003, with Lorraine Ramig.
- Using a Portable Computing Device to Assist Persons with Parkinson Disease. Coleman Institute ($25,000), 2001-2002, with Lorraine Ramig.
- The Bifrost Location Independent Computing System. Microsoft Corporation ($185,000), 2000-2001, with Evan Speight, Cornell University.
- Optimizing VHDL Intermediate Form. DARPA ($751,000), 1997-2001, with Keith Cooper and Linda Torczon.
- Efficient Shared Memory Parallel Computing on Personal Computer Clusters. Compaq Computer Corporation, ($260,000), 1998-2000.
- Efficient Shared Memory Parallel Computing on Personal Computer Clusters. Texas Advanced Research Program / ATP ($203,749), 1998-2000.
- Support for the Brazos Project. Intel Corporation ($14,060), 1999.
- Support for the Brazos Distributed Shared Memory Project. Microsoft Corporation ($50,000), 1998-1999.
- Support for the Brazos Project. Intel Corporation ($38,880), 1998.
- Support for the Brazos Distributed Shared Memory Project. Tandem Corporation ($30,000), 1998.
- Distributed Shared Memory Research. Microsoft Corporation ($18,000), 1997-1998.
- A Medium Scale, Tightly Coupled, Shared-Memory Multiprocessor. NSF CISE Research Instrumentation ($82,932), with Sarita Adve et al., 1996.
- Multiprocessor Cluster Computing. NSF ($1,050,000), with Willy Zwaenepoel et al., 1995-2000.
- Physical Properties of Latex Products. Ansell Medical ($50,000), 1995-1996
- Performance Debugging of Shared Memory Parallel Programs, with Sarita Adve. Texas Advanced Research Program / ATP ($207,514), 1994-1996.
- Efficient Parallel Computing on Networks of Personal Computers, with Willy Zwaenepoel. Texas Advanced Research Program / ATP ($480,000), 1994-1996.
- Physical Properties of Latex Surgical Gloves. Regent Hospital Products, Ltd. ($42,500), 1994-1995
- Management of Mobility-Related Data Using Shared-Memory Techniques. Bell Northern Research ($90,000), 1992-1995.
- Physical Properties of Latex Surgical Gloves. Smith and Nephew Perry ($50,000), 1993-1995
- A Software Environment for a Scalable Shared Memory Processor. NASA AMES ($66,000), 1991-1994.
- Parallel Fault-Tolerant Robot Control. NASA JSC ($66,000), 1992-1995.

- The Interaction of Architecture and Operating System in the Design of a Scalable Shared Memory Multiprocessor. IBM ($38,500), 1991-1993.
- Texas Geophysical Parallel Computation Project. State of Texas, PI: Ken Kennedy. ($100,000), 1990-1992.
- Research Experience for Undergraduates. NSF ($4,400), 1991-1993.
- Adaptive Software Cache Management for Distributed Shared Memory Architectures. NSF Research Initiation Award ($60,000), 1990-1993.
- Special Graduate Student Education and Research Award. (NSF, $6,000), 1990-1992.

Education Funding

- Support for the Rural Engineering Program. Daniels Fund ($15,000), 2007.
- Support for the Rural Engineering Program. Shell Corporation ($15,000), 2007.
- Support for the Rural Engineering Program. GE Foundation ($100,000), 2006.
- Support for the Rural Engineering Program. Lockheed Martin Corporation ($15,000), 2006.
- Support for the Rural Engineering Program. Microsoft Corporation ($20,000), 2005.
- Support for the Rural Engineering Program. Bechtel Foundation ($150,000), 2005-2009.
- Rummel Creek Elementary University: A K-5 Institution of Higher Learning, with Shirley Lincoln. R.J.R. Reynolds Next Century School Fund ($750,000), 1992-1995.
- Viewlogic Systems Corporate Grant. ($846,000), 1990-1995.
- Enhancement of the Rice Computer Systems and Digital Design Laboratories. Hewlett-Packard Company Corporate Grant ($62,205), 1989.

Professional Activities

- Program Committee Regional Chair (North America), 2009 IEEE/ACM International Conference on Information and Communication Technologies and Development (ICTD2009)
- President of the Governing Board, Engineers Without Borders-USA (2006-2009)
- Member of the Board, National Center for Women and IT (2007-)
- Member of the Academic Alliance, National Center for Women and IT (2004-)
- Member of the Governing Board, Engineers Without Borders-USA (2003-2009)
- Program Committee Member, 2007 IEEE/ACM International Conference on Information and Communication Technologies and Development (ICTD2007)
- Program Committee Member, 2004 Assets: The Sixth International ACM SIGCAPH Conference on Assistive Technologies
- Program Committee Member, 2002 Assets: The Fifth International ACM SIGCAPH Conference on Assistive Technologies
- Program Committee Member, 2000 International Conference on High Performance Parallel Computing
- Program Committee Member, 1998 International Conference on Parallel Processing
- Program Committee Member, 1998 Windows NT Research Symposium
- Program Committee Member, 1996 Symposium on Architectural Support for Programming Languages and Operating Systems (ASPLOS)
- Program Committee Member, 1996 International Conference on Distributed Computing Systems
- General Chair, 1995 Symposium on Operating Systems Principles
- Referee for:

  ACM Transactions on Computer Systems, IEEE Transactions on Computers, IEEE Transactions on Parallel and Distributed Systems, Journal of Parallel and Distributed Computing, Software Practice and Experience, IEEE Computer Magazine, Conference on Object Oriented Programming, Systems, Languages, and Applications, International Symposium on Computer Architecture, Symposium on High Performance Computer

Architecture, Symposium on Operating Systems Principles, International Conference on Architectural Support for Programming Languages and Operating Systems, Hawaii International Conference on System Sciences, Symposium on the Frontiers of Massively Parallel Computation, NSF, and NSERC.

Honors and Awards

- Awarded Archuleta Endowed Professorship (2007)
- Elected to membership in Eta Kappa Nu (2007)
- Fellow of the ATLAS Institute (2005)
- European Academy of Sciences (2003)
- W.M. Keck Foundation National Award for Engineering Teaching Excellence (1996)
- George R. Brown Award for Innovative Teaching (1995)

Courses Taught

At The University of Colorado:

- CSCI/ATLS 1220 – Virtual Worlds: Introduction to Computer Science
- GEEN 1400 -Introduction to Engineering Design
- ATLS 2000 – The Meaning of Information Technology
- CSCI 2830 – The Profession of Computer Science
- CSCI 5573 - Advanced Operating Systems
- ECEN 4633/5633 – Hybrid Embedded Systems
- CSCI 7000 - Advanced Topics: Mobile Computing
- CSCI 7000 - Advanced Topics: ATLAS Interdisciplinary Graduate Seminar

At Rice University:

- ELEC 201 - Introduction to Engineering Design
- ELEC 424 - Computer System Design
- ELEC/COMP 425 - Computer Systems Architecture
- ELEC 426 - Digital System Design
- ELEC/COMP 525 - Advanced Computer Architecture
- ELEC 625 - High Performance Microprocessor Design
- ELEC 693 - Advanced Topics: Cluster Multiprocessors
- ELEC 693 - Advanced Topics: Operating System/Architecture Interaction
- ELEC 693 - Advanced Topics: Shared Memory Multiprocessing
- ELEC 693 - Advanced Topics: Techniques for Tolerating Memory Latency
- ELEC 694 - Advanced Topics: Issues in Parallel Computing
- ELEC 694 - Advanced Topics: Shared Memory Parallel Programming

University Activities and Service

At The University of Colorado:

- Flagship 2030 Facilities Task Force (2008)
- Law School Promotion Advisory Committee (2006)
- Provost's Assessment Committee (2004-), Co-chair (2005)
- Chair, Undergraduate Education Council (2002-)
- Chair, Graduate Education Council (2002-)
- Engineering Dean Search Committee (2000-2002)
- Academic Affairs Budget Advisory Committee (2001-2002)

- Faculty Teaching Excellence Program New Faculty Mentor (2001-2002)
- Faculty Teaching Excellence Program Summer Institute Steering Committee (2001-2002)
- Chair, Computer Science Search Committee (2005-2006)
- Chair, Computer Science Affiliates Committee (2000-2002)
- CS Department Executive Committee (2000-2002, 2004-2006, 2007)
- CS Department Undergraduate Curriculum Committee (2000-2002)
- CS Department Resource Committee (2000-2002)
- CS Department RI Grant Steering Committee (2000-2002)
- Faculty supervisor for 28 undergraduate independent study projects

At Rice University:

- University Alcoholic Beverage Policy Advisory Committee (1996-2000), Chair (1997-2000)
- New College Construction User Liaison (1998-2000)
- Chair, New College Planning Committee (1998-2000)
- Chair, New College Populating Committee (1998-2000)
- Chair, Ad Hoc Thresher Advisory Committee (1997-1998)
- Jones School of Management Strategic Planning Steering Committee (1996-1997)
- University Undergraduate Curriculum Committee (1996-1997)
- University Crisis Management Team (1996-1998)
- Faculty Sponsor, Women's Lacrosse (1997-2000)
- Faculty Sponsor, Latter Day Saints Student Association (1997-2000)
- Faculty Sponsor, Women's Rugby (1995-2000)
- Faculty Sponsor, Men's Rugby (1998-2000)
- Resident Faculty Associate, Wiess College (Spring 1988)
- Faculty Associate, Wiess College (1988-2000) (Outstanding Associate in 1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996)
- Engineering Divisional Advisor, Wiess College (1989-1996)
- Computer Systems Engineering Undergraduate Advisor (1989-1990)
- Computer Systems Engineering Graduate Advisor (1990-1995)
- Jones School of Management Friedkin Chair Search Committee (1994-1996)
- University Computer Committee (1988-1993)
- University Committee on Community Affairs (1993-1994)
- ECE Department Colloquia Chairman (1988-1990)
- ECE Department Undergraduate Committee (1989-1990)
- ECE Department Graduate Committee (1990-1995,1996-1997), Chairman (1991-1992, 1994-1995)
- ECE Department Comprehensive Qualifying Exam Committee (1990-1995,1996), Chairman (1991-1992, 1994-1995)
- ECE Department Computer Engineering Faculty Recruiting Chairman (1991-1994)
- ECE Department Computer Committee (1992-1995)
- ECE Department Corporate Affiliates Committee (1990-1995), Chairman (1994-1996)
- Engineering Facilities Committee (1988-1995)
- MBA / Professional Masters Joint Degree Planning Committee (1989)
- Faculty supervisor for 71 undergraduate independent study projects

Theses Supervised

1. S. Revi Sterling. Advancement Through Interactive Radio. Ph.D. Dissertation. December, 2008.
2. Joseph Dunn. Secure Sharing Across Trust Boundaries. PhD Dissertation. August, 2007.
3. Brian Shucker. Control of Distributed Robotic Macrosensors. Ph.D. Dissertation. July, 2006.

4. L. Daniel Crawl. Affinity Directed Mobility. Ph.D. Dissertation. July, 2006.
5. Carlos Matos. Adapting a PDA to Enhance Voice Treatment in Parkinsons Disease. Masters thesis. December, 2001.
6. Hazim Abdel-Shafi. Reliable Parallel Computing. Ph.D. Dissertation. May, 2000.
7. Damian Dobric. Efficient User-Level Communication. Masters thesis. May, 2000.
8. Vasilios Balabanos. Optimization of Logic Synthesis. Masters thesis. May, 2000.
9. W. Evan Speight. Efficient Runtime Support for Cluster-Based Distributed Shared Memory Multiprocessors. Ph.D. Dissertation. July, 1997.
10. Yanyang Xiao. Memory Architecture In Multi-channel Optically Interconnected Distributed Shared Memory Multiprocessor Systems. Masters thesis. July, 1996.
11. Dierdre Hamilton. Effectiveness and Performance Analysis of a Class of Parallel Robot Controllers with Fault Tolerance. Ph.D. Dissertation. May, 1996. (jointly supervised with Ian Walker)
12. Rajat Mukherjee. The Interaction of Architecture and Operating System in the Design of a Scalable Shared Memory Multiprocessor. Ph.D. Dissertation. October, 1994.
13. Kenneth Mark Maxham. ASPEN: High-Performance Hardware Support for Distributed Shared-Memory. Masters thesis. May, 1994.
14. W. Evan Speight. ParaView: Performance Debugging Through Visualization of Shared Data. Masters thesis. August, 1993.
15. John Carter. Munin: Efficient Distributed Shared Memory Based On Multi-Protocol Release Consistency. Ph.D. Dissertation. December, 1992. (jointly supervised with Willy Zwaenepoel)
16. Dierdre Hamilton. Performance and Reliability of a Parallel Robot Controller. Masters thesis. May, 1992. (jointly supervised with Ian Walker)
17. Jay Greenwood. The Design of a Scalable, Hierarchical-Bus, Shared-Memory Multiprocessor. Masters thesis. May, 1992.
18. Rajat Mukherjee. Simulation of Shared Memory Parallel Systems. Masters thesis. May, 1990.

Patents

1. Apparatus for Electrical Connection of Glove Monitor to Patient, U.S. Patent No. 5,658,277, August 19, 1997.
2. Enhanced Monitoring Device for Surgical Gloves and Other Barriers, U.S. Patent No. 5,389,097, February 14, 1995.

Publications

1. S. Revi Sterling, Sophia Huyer, John K. Bennett. “89.1 FM: The Place for Development: Power shifts and participatory spaces in ICTD.” *Community Informatics*, To Appear, 2009.
2. A.E. Halpern, S. Sapir, L. O. Ramig, C. Matos, J. Petska, J. Spielman, J. K. Bennett, and J. M. Pogoda. Intensive Voice Treatment (LSVT/LOUD) for Parkinson Disease Supported by Technology. Submitted to *Journal of Speech, Language, and Hearing Research*.
3. S. Revi Sterling, John W. O'Brien and John K. Bennett. Advancement Through Interactive Radio. In *Communication for Social Change*. Information Systems Frontiers. 11(2), 2009.
4. Brian Shucker, Todd Murphey, and John K. Bennett. Convergence Preserving Switching for Topology Dependent Decentralized Systems. *IEEE Transactions on Robotics.* 24(6), 1405-1415, Dec. 2008.
5. S. Revi Sterling and John K. Bennett. User Centric Design for Innovative Women. *Information for Development*, 3(2), 49-51, July, 2008.
6. S. Revi Sterling, John W. O’Brien and John K. Bennett. Advancement Through Interactive Radio. In *Proceedings of the 2nd IEEE/ACM International Conference on Information and Communication Technologies and Development (ICTD 2007).* Bangalore, India. December 2007.
7. John Giacomoni, John K. Bennett, Antonio Carzaniga, Douglas C. Sicker, Manish Vachharajani, Alexander L. Wolf. Frame shared memory: Line-rate Networking on Commodity Hardware. In

Proceedings of ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS 2007). Coronado Springs Resort, Orlando, Florida, USA. December 2007

8. S. Revi Sterling and John K. Bennett. Hear My Voice: A System to Capture Personal Ethnographies from African Mwethyas. In *Proceedings of the Third International Ethnographic Praxis in Industry Conference (EPIC 2007)*. Keystone, Colorado, USA. October 2007.
9. S. Revi Sterling and John K. Bennett. Broadcasting women: Participatory Community Radio as a Space of Empowerment. In *Proceedings of the Royal Geographic Society Annual International Conference (RGS 2007).* London, Great Britain. August 2007.
10. S. Revi Sterling and John K. Bennett. Out of the Margins: Lessons from Africa on Including Women in Community Radio. In *Proceedings of Community Radio India 2007 Conference (eIndia 2007).* New Delhi, India. July 2007.
11. Brian Shucker, Todd Murphey, and John K. Bennett. Switching Rules for Decentralized Control with Simple Control Laws. In *Proceedings of the 2007 American Control Conference (ACS 2007).* New York City, USA, July 2007.
12. Brian Shucker, Todd Murphey, and John K. Bennett. Testbed Implementation of Wireless Distributed Control. In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation (ICRA 2007)*. Rome, Italy. April 2007.
13. S. Revi Sterling and J.K. Bennett. Broadcasting Women, Broadcasting Resistance. In *Proceedings of the Society for Social Studies of Science* (4S) Vancouver, BC, November 2006.
14. Daniel Crawl, Joseph Dunn, Avneesh Bhatnagar, Evan Speight, and John Bennett. Using Location Dependence to Manage Mobile Data. In *Proceedings of the 3rd Annual International Conference on Mobile and Ubiquitous Systems: Networks and Services* (MOBIQUITOUS 2006), San Jose, California. July, 2006.
15. S.R. Sterling and J.K. Bennett. Women's Identity Production Through Participatory Community Radio. In *Proceedings of the International Conference on Communication for Development and Social Change.* July, 2006.
16. Brian Shucker, Todd Murphey, and John K. Bennett. Switching Control Without Nearest Neighbor Rules. In *Proceedings of the 2006 American Control Conference.* 2006.
17. Brian Shucker and John K. Bennett. Cooperative Control Using Occasional Non-Local Interactions. In *Proceedings of 2006 IEEE International Conference on Robotics and Automation.* June, 2006.
18. Graham Schelle, Daniel Fay, Dirk Grunwald, Dan Connors and John Bennett. An Evolving Curriculum to Match the Evolution of Reconfigurable Computing Platforms. In *Proceedings of WRCE2006 (IEEE Computer Society Workshop on Reconfigurable Computing Education)*. March, 2006.
19. Brian Shucker and John K. Bennett. Target Tracking with Distributed Robotic Macrosensors. In *Proceedings of Milcom 2005* (IEEE/AFCEA Military Communications Conference), Oct. 2005, Atlantic City NJ.
20. J. Dunn, M. Neufeld, A. Sheth, D. Grunwald and J. Bennett. A Practical Cross-Layer Mechanism for Fairness in 802.11 Networks. *ACM/Kluwer Mobile Networks and Applications Journal.* December, 2005.
21. T. Mayes and J.K. Bennett. A Survey of Current ABET Best Practices. *Proceedings of the 2005 ASEE Annual Conference.* (ASEE 2005). June 2005. Portland, Oregon.
22. J. Dunn, M. Neufeld, A. Sheth, D. Grunwald and J. Bennett. A Practical Cross-Layer Mechanism for Fairness in 802.11 Networks. *Proceedings of the First Annual International Conference on Broadband Networks* (BroadNets 2004). October 2004. San Jose, California.
23. B. Shucker and J. Bennett. Scalable Control of Distributed Robotic Macrosensors. *Proceedings of the 7th International Symposium on Distributed Autonomous Robotic Systems* (DARS'04). June 2004. Toulouse, France
24. H. Abdel-Shafi, W.E. Speight and J.K. Bennett. Raptor: Integrating checkpoints and thread migration for cluster management. In *Proceedings of the 22nd Symposium on Reliable Distributed Computing (SRDS2003)*, 2003.

25. A. Bhatnagar, E. Speight, D. Crawl, J. Dunn, and J. K. Bennett. "Application Management Techniques for the Bifrost System." In *Proceedings of the Fifth IEEE Workshop on Mobile Computing Systems & Applications*, October, 2003.
26. W.E. Speight, H. Abdel-Shafi, and J.K. Bennett. WSDLite: A lightweight alternative to Windows Sockets Direct Path. In *Proceedings of the 2000 USENIX Windows Systems Research Symposium*, August, 2000.
27. H. Abdel-Shafi, W.E. Speight and J.K. Bennett. Efficient user-level thread migration and checkpointing on Windows NT clusters. In *Proceedings of the 1999 USENIX Windows/NT Research Symposium*, 1-10, July, 1999.
28. W.E. Speight, H. Abdel-Shafi, and J.K. Bennett. Realizing the performance potential of the Virtual Interface Architecture. In *Proceedings of the 13th ACM International Conference on Supercomputing (ICS)*, June 1999.
29. D.L. Hamilton, I.D. Walker, and J.K. Bennett. Parallel robot control using speculative computation. In *Journal of Robotics and Automation*, 13(4), 101-112, Dec, 1998.
30. W.E. Speight, H. Abdel-Shafi, and J.K. Bennett. An integrated shared-memory/message passing API for cluster-based multicomputing. In Proceedings *of the Second International Conference on Parallel and Distributed Computing and Networks*, December, 1998.
31. J.R. Nelson, T.A. Roming, J.K. Bennett. A whole glove method for evaluation of surgical gloves as barrier to virus. *Journal of Allergies and Clinical Immunology*, 1998.
32. W.E. Speight and J.K. Bennett. Using multicast and multithreading to reduce communication in software DSM systems. In *Proceedings of the Fourth Symposium on High Performance Architecture (HPCA)*, 312-323, February, 1998.
33. W.E. Speight and J.K. Bennett. Brazos: A third generation DSM system. In *Proceedings of the 1997 USENIX Windows/NT Workshop*, 95-106, August, 1997.
34. D.L. Hamilton, I.D. Walker, and J.K. Bennett. Parallel robot control using speculative computation. In *Journal of Robotics Systems*, 1997.
35. J.K. Bennett. The clinical significance of hydration in natural rubber latex gloves. In *Surgical Services Management*, 3(2), 29-33, Feb, 1997.
36. D.L. Hamilton, I.D. Walker, and J.K. Bennett. Fault tolerance versus performance metrics for robot manipulators. In *Reliability Engineering and System Safety*, 53(1996), 309-318, 1996.
37. J.K. Bennett, K.E. Fletcher, and W.E. Speight. The performance value of shared network caches in clustered multiprocessor workstations. In *Proceedings of the 16th International Conference on Distributed Computing Systems (ICDCS-16)*, May 1996.
38. Yanyang Xiao and J.K. Bennett. Memory organization in multi-channel optical networks: NUMA and COMA revisited. In *Proceedings of the 10th ACM International Conference on Supercomputing (ICS)*, May 1996.
39. D.L. Hamilton, I.D. Walker, and J.K. Bennett. Fault tolerance versus performance metrics for robot manipulators. *In Proceedings of the 1996 International Conference on Robotics and Automation*, 3073-3080, Minneapolis, MN, 1996.
40. D.L. Hamilton, I.D. Walker, and J.K. Bennett. Parallel robot control using speculative computation. In *Proceedings of the 1996 International Conference on Robotics and Automation*, 3420-3427, Minneapolis, MN, 1996.
41. J.B. Carter, J.K. Bennett, and W. Zwaenepoel. Techniques for reducing consistency-related communications in distributed shared memory systems. *ACM Transactions on Computers*, 13(3), 205-243, Aug. 1995.
42. J.K. Bennett, J.B. Carter, and W. Zwaenepoel. Adaptive software cache management for distributed shared memory architectures. In *The Cache Coherence Problem in Shared Memory Multiprocessors: Software Solutions*, Igor Tartalja and Veljko Milutinovic, editors, IEEE Computer Society Press, 1995.
43. J.K. Bennett. The feasibility of using electrical means for monitoring barrier integrity in natural rubber latex gloves. In *Journal of Infection Control and Hospital Epidemiology*, 1995.

44. R. Mukherjee and J.K. Bennett. Operating system design principles for scalable shared memory multiprocessors. In *Proceedings of the Symposium on Parallel and Distributed Computing Systems (PDCS-95)*, 1995.
45. R. Mukherjee, J.K. Bennett, and J.A. Greenwood. The effects of architecture on the performance of latency hiding via rapid context switching. In *Proceedings of the 7th IASTED International Conference on Parallel and Distributed Computing and Systems*, 1995.
46. J.K. Bennett. The feasibility of using electrical means for monitoring barrier integrity in natural rubber latex gloves. In *Proceedings of the American College of Surgeons / Center for Disease Control Conference on Bloodborne Pathogens*, February, 1994.
47. D.L. Hamilton, M.L. Visinsky, J.K. Bennett, J.R. Cavallaro, and I.D. Walker. Fault tolerant algorithms and architectures for robotics. In *Proceedings of the 1994 Mediterranean Electrotechnical Conference*, 1034-1036, Antalya, Turkey, April 1994.
48. J.K. Bennett. Hydration and conductivity in natural rubber latex gloves. *Source to Surgery* 1(3), 1-4, October, 1993.
49. J.K. Bennett. Investigating the physical properties of latex gloves. *In Touch* 2(2), 1-2, May 1993.
50. J.K. Bennett, S. Dwarkadas, J.A. Greenwood, and Evan Speight. Willow: A scalable shared memory multiprocessor. In *Proceedings of SuperComputing 92*, IEEE Computer Society Press, pp. 336-345, Nov. 1992.
51. D.L. Hamilton, J.K. Bennett, and I.D. Walker. Simulation of a reliable robot control architecture. In *Proceedings of the 1992 International Simulation Technology Conference*, 321-327, Nov. 1992.
52. D.L. Hamilton, J.K. Bennett, and I.D. Walker. Parallel fault-tolerant robot control. In *Proceedings of the 1992 SPIE Conference on Cooperative Intelligent Robotics in Space III*, pp. 251-261, Nov. 1992.
53. J.K. Bennett, J.B. Carter, A.L. Cox, E.N. Elnozahy, D.B. Johnson, P. Keleher and W. Zwaenepoel. Distributed shared memory: Experience with Munin. *In Proceedings of the Fifth European ACM SIGOPS Workshop*, July, 1992.
54. J.B. Carter, J.K. Bennett, and W. Zwaenepoel. Implementation and performance of Munin. In *Proceedings of the 13th Symposium on Operating System Principles*, pp. 152-164, Oct. 1991.
55. J.K. Bennett, J.B. Carter, and W. Zwaenepoel. Toward large scale shared memory multi-processing. In *Scalable Shared Memory Multiprocessors*, M. Dubois and Shreekant Thakkar, editors, Kluwer Academic Publishers, Nov. 1991.
56. J.K. Bennett, J.B. Carter, and W. Zwaenepoel. Munin: Distributed shared memory using multi-protocol release consistency, In *Operating Systems of the 90s and Beyond*, A.I. Karshner and J. Nehmer, editors, Lecture Notes in Computer Science, Springer-Verlag LNCS 563, pp. 56-60, 1991.
57. J.K. Bennett, J.B. Carter, and W. Zwaenepoel. Adaptive software cache management for distributed shared memory architectures. In *Proceedings of the 17th International Symposium on Computer Architecture*, 125-134, May 1990.
58. J.K. Bennett, J.B. Carter, and W. Zwaenepoel. Munin: Distributed shared memory based on type-specific memory coherence. In *Proceedings of the Second ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP),* pp. 168-176, Mar. 1990.
59. J.K. Bennett. Experience with Distributed Smalltalk. *Software Practice and Experience*, 20(2), 157-180, Feb. 1990.
60. R. Mukherjee and J.K. Bennett. Simulation of parallel programs on a shared memory multiprocessor. In *Proceedings of the Hawaii International Conference on System Sciences (HICSS23)*, pp. 242-251, Dec. 1989.
61. J.K. Bennett. The design and implementation of Distributed Smalltalk. In *Proceedings of the Second Annual Conference on Object-Oriented Programming Systems, Languages, and Applications*, 318-330, Orlando, Florida, Oct. 1987.
62. J.K. Bennett. Raster operations: integrating text and graphics in high performance systems. *Byte*, 10(12): 1187-1192, Nov. 1985.

Appendix B: Materials Considered

B.1 Bates Stamped Documents

B0547811-0547812
BAR171664
BAR191458 – 191459
BAR191460 – 191463
BAR191485 – 191486
BOR 007479-007480
C-12462 – 12463
C-12471
DELL00868 - 0086873
FL AG 0005434-0005439
FL AG 0010974-0010981
FL AG 0030494-0030502
FL AG 0047255-0047323
FL AG 0077490-0077522
FL AG 0080063-0080064
FL AG 0080501-0080504
FL AG 0088711-0088719
FL AG 0094344-0094412
FL AG 0097685
FL AG 0098424-0098425
FL AG 0100202-0100246
FL AG 0100492-0100519
FL AG 0100530-0100533
FL AG 0100534-0100537
FL AG 0100539-:0100543
FL AG 0100749-0100750
FL AG 0102980-0102985
FL AG 0102987-0103004
FL AG 0103212-0103221
FL AG 0106458-0106463
IBM 0410332099-0410332199
IBM 7510051005
IBM 7510090100-7510090101
IBM 7510131528-7510131530
IBM 7510131531-7510131532
IBM 7510131533-7510131535
IBM 7510131539-7510131540
IBM 7510131539-7510131540
IBM 7510131560-7510131561
IBM 7510131560-7510131561
IBM 7510131564 -7510131565
IBM 7510138374-7510138377
IBM 7510138374-7510138379
IBM 7510138378-7510138379
IBM 7510138380-7510138381
IBM 7510138383
IBM 7510172810-7510172830
IBM 7510215430-7510215485
IBM 7510241168-7510241169
IBM 7510250263
IBM 7510250333
IBM 7510250349-7510250357
IBM 7510250608-7510250610
IBM 7510251203-7510251218
IBM 7510251219-7510251261
IBM 7510251292-7510251298
IBM 7510251292-7510251320
IBM 7510251299-7510251320
IBM 7510251324-7510251328
IBM 7510251331-7510251334
IBM 7510251351-7510251352
IBM 7510251353
IBM 7510251895
IBM 7510251896
IBM 7510251955-7510251956
IBM 7510251964-7510251970
IBM 7510251971-7510251972
IBM 7510251973-7510251974
IBM 7510279474-7510279498
IBM 7510279757-7510279767
IBM 7510295373
IBM 7510341143-7510341146
IBM 7510341168-7510341169
IBM 7510346947-7510346991
IBM 7510366183
IBM 7510371315-7510137318
IBM 7510372113-7510372116
IBM 7510374019-7510374021

IBM 7510374577-7510374580
M 1010467-1010468
M 1010506-1010508
M 1016006-1016012
M 1020718-1020728
M 1022647-1022653
M 1027348-1027350
M 1028049-1028057
M 1028272-1028280
MS 0073067-0073069
MS 0073148-0073154
MS 0079806-0079807
MS 0083693-0083722
MS 0083723-0083809
MS 0084064-0084071
MS 0084071-0084101
MS 0084179-0084221
MS 0091921-0091923
MS 0092305-0092307
MS 0092403-0092404
MS 0096813-0096921
MS 0097113-0097126
MS 0097121-0097126
MS 0103187-0103196
MS 0114639-0114660-2x
MS 0119827-0119832
MS 0120100-0120113
MS 0120221-0120225
MS 0122253-0122257
MS 0127353-0127355
MS 0137564-0137567
MS 0145793-0145799
MS 0150224-0150225
MS 0150226-0150229
MS 0150236-0150238
MS 0150261-0150265
MS 0150275-0150278
MS 0150287-0150292
MS 0150299-1263310
MS 0150317-0150333
MS 0150327-0153063
MS 0150654-0150655
MS 0150656
MS 0150711
MS 0150722-0150723
MS 0150781
MS 0150887-0150889
MS 0151082
MS 0152183-0152189
MS 0153097-0153098
MS 0153361-0153362
MS 0153730-0153740
MS 0154480-0154484
MS 0154956-0154960
MS 0155114-0155120
MS 0155166-0155167
MS 0156210-0156211
MS 0170968-0171120
MS 0171828-0171829
MS 0182883-0182885
MS 0182886-0182887
MS 0183031-0183032
MS 0183137-0183138
MS 0184621
MS 0184702-0184708
MS 0184859-0184860
MS 0184894
MS 0185379-0185380
MS 0185836-0185837
MS 0185844-0185885
MS 0185857
MS 0185866-0185868
MS 0185884-0185885
MS 0186262-0186263
MS 0186379-0186380
MS 0186399
MS 0186416
MS 0186619-018620
MS 039037-039039
MS 039089-039091
MS 044118-044120

MS 045558-006559
MS 045558-045559
MS 049065-049067
MS 060468-060470
MS 060468-060474
MS 3171073-3171074
MS 5011634-5011649
MS 5014820
MS 5019485-5019486
MS 5025058
MS 5025063
MS 5025271
MS 5025330
MS 5025375-5025376
MS 5025386
MS 5029173-5029176
MS 5031401-5031403
MS 5031554
MS 5031555
MS 5031898-5031900
MS 5033031-5033033
MS 5033637-5033639
MS 5034510-5034511
MS 5035892-5035894
MS 5035972-5035974
MS 5035975-5035976
MS 5036025-5036030
MS 5036251-5036252
MS 5036284
MS 5036419
MS 5036507-5036508
MS 5036518-5036519
MS 5037753-5037755
MS 5039037-5039039
MS 5039088
MS 5039089-5039091
MS 5039718-5039719
MS 5039792-5039794
MS 5041080-5041082
MS 5041425-5041427
MS 5041454-5041456
MS 5042220-5042222
MS 5042223
MS 5042229
MS 5043511-5043513
MS 5043525-5043526
MS 5045670
MS 5046397-5046414
MS 5048442-5048452
MS 5048630-5048636
MS 5049163
MS 5055390
MS 5056342
MS 5058284-5058287
MS 5059922
MS 5059924
MS 5059935
MS 5060417
MS 5060540
MS 5060656
MS 5060672-5060673
MS 5060675-5060677
MS 5060870
MS 5061001
MS 5061030
MS 5063860-5063862
MS 5064010-5064011
MS 5064050-5064051
MS 5064058
MS 5065145-5065148
MS 7013213-7013224
MS 7041504-7041505
MS 7042841-7042848
MS 7043091-7043097
MS 7043109-7043126
MS 7045839-7045842
MS 7048981-7048991
MS 7049089-7049100
MS 7051073-7051079
MS 7058499
MS 7058541-7058561
MS 7059112

MS 7066239-7066244
MS 7073039-7073046
MS 7079459-7079461
MS 7080372-7080373
MS 7080424
MS 7080438-7080440
MS 7080463-7080465
MS 7080466-7080468
MS 7080472-7080474
MS 7080485-7080487
MS 7080511-7080512
MS 7080513-7080515
MS 7080520
MS 7080546-7080548
MS 7080601-7080603
MS 7080895-7080897
MS 7081613-7081615
MS 7082429-7082430
MS 7082447-7082451
MS 7082580-7082582
MS 7083578
MS 7083975-7083976
MS 7084748-7084749
MS 7085229-7085230
MS 7085669
MS 7085723
MS 7085784
MS 7085793-7085795
MS 7085836-7085837
MS 7086397
MS 7086399
MS 7086411-7086412
MS 7086438-7086440
MS 7086445
MS 7086459
MS 7086482-7086483
MS 7086507-086508
MS 7086583-7086584
MS 7086900-7086901
MS 7087101-7087103
MS 7087512
MS 7087784-7078785
MS 7087824
MS 7087831
MS 7087847-7087850
MS 7087936-7087937
MS 7087944-7087945
MS 7088886-7088894
MS 7088935-7088938
MS 7089438-7089442
MS 7091860-7091861
MS 7092088-7092090
MS 7092256-7092257
MS 7092351-7092353
MS 7092476
MS 7093034
MS 7093039-7093041
MS 7093049
MS 7093119
MS 7093163
MS 7094219-7094220
MS 7094221-7094222
MS 7094235-7094236
MS 7094240
MS 7094453-7094454
MS 7094455-7094456
MS 7094461
MS 7094469-7094470
MS 7094492-7094494
MS 7094721-7094731
MS 7096165-7096169
MS 0076890-0076892
MS 66008820
MS 7004127-004147
MS 7004216-004217
MS 7041504
MS98 0009584-0009598
MS98 0102394-0102410
MS98 0103242
MS98 0103243
MS98 0103258-0103260
MS98 0103692

MS98 0104677
MS98 0104679-0104691
MS98 0104692-0104693
MS98 0105322-0105323
MS98 0108494
MS98 0113095-0113103
MS98 0118760-0118788
MS98 0119267-0119268
MS98 0120900-0120902
MS98 0126146-0126149
MS98 0151218-0151221
MS98 0168612-0168615
MS98 0185400-0185405
MSB 0029580-0029581
MSC 00114461.01-00114461.55
MSC 00511224
MSC 00513135-00513154
MSC 00536472-00536473
MSC 00544224
MSC 00565157-00565172
MSC 00568050-00568067
MSC 00613135-01154564
MSC 00623987-00624391
MSC 00624313-00624361
MSC 00700613-00700618
MSC 00718944-00718954
MSC 00718959-00718975
MSC 00746547
MSC 00762731-00762998
MSC 00795586
MS-CCPMDL 000000187255-000000187259
MS-CCPMDL 000000271374-000000271486
MS-CCPMDL 000000364423-000000364436
MS-CCPMDL 000000364437-000000364476
MS-CCPMDL 000000364509-000000364510
MS-CCPMDL 000000364513-000000364520
MS-CCPMDL 000000364521-000000364526
MS-CCPMDL 000000365093-000000365118
MS-CCPMDL 000000365153
MS-CCPMDL 000000365162
MS-CCPMDL 000000365209
MS-CCPMDL 000000365215
MS-CCPMDL 000000379829-000000379861
MS-CCPMDL 000000392736-000000392751
MS-CCPMDL 000000411895-000000411896
MS-CCPMDL 000000443572
MS-CCPMDL 000000500504-000000500558
MS-CCPMDL 000005006716
MS-CC-RN 000000059375-000000059380
MS-CC-RN 000000695358-000000695363
MS-CC-Sun 000001273156-000001273164
MS-CC-Sun 000001409566-000001409601
MS-CC-Sun 000001526107-000001526117
MS-CC-Sun 000001589242-000001589244
MS-CC-Sun 000001742530-000001742532
MS-L 010831
MS-PCA 1001432-1001433
MS-PCA 1001461
MS-PCA 1001481
MS-PCA 1001568-1001574
MS-PCA 1001613-1001643
MS-PCA 1001781-1001783
MS-PCA 1008273-1008320
MS-PCA 1034333-1034388
MS-PCA 1035541-1035545
MS-PCA 1051170-1051332
MS-PCA 1084435-1084438

MS-PCA 1084903
MS-PCA 1084966-1085014
MS-PCA 1102674-1102675
MS-PCA 1182046-1182047
MS-PCA 1242563-1242566
MS-PCA 1254189-1254200
MS-PCA 1301178-1301181
MS-PCA 1314286-1314306
MS-PCA 1384929-1384952
MS-PCA 1384999-1385010
MS-PCA 1389717
MS-PCA 1391451-1391467
MS-PCA 1396725
MS-PCA 1402244-1402248
MS-PCA 1402253-1402256
MS-PCA 1405389-01434191
MS-PCA 1405389-1405390
MS-PCA 1405588-1405593
MS-PCA 1419058-1419060
MS-PCA 1438869-1438871
MS-PCA 1474638-1474647
MS-PCA 1505251-1505252
MS-PCA 1541686-1541689
MS-PCA 1542047-1542048
MS-PCA 1566791-1566797
MS-PCA 1566798-1566807
MS-PCA 1566798-1567807
MS-PCA 1576429-1576431
MS-PCA 1579932
MS-PCA 1619544-1619549
MS-PCA 1620115-1620118
MS-PCA 1624111-1624146
MS-PCA 1624113-1624145
MS-PCA 1653226-1653271
MS-PCA 1714801-1714806
MS-PCA 1730676-1730678
MS-PCA 1730891-1730893
MS-PCA 1816094-1816096
MS-PCA 2013978-2013981
MS-PCA 2061739-2061742
MS-PCA 2107155-2107156
MS-PCA 2212583-2212633
MS-PCA 2404815-2404818
MS-PCA 2405055-2405058
MS-PCA 2405070-2405072
MS-PCA 2405073-2405075
MS-PCA 2405076-2405079
MS-PCA 2405080-2405083
MS-PCA 2405084-2405088
MS-PCA 2405089-2405093
MS-PCA 2405196-2405198
MS-PCA 2409774-2409775
MS-PCA 2409907-2409908
MS-PCA 2412200-2412201
MS-PCA 2535283-2535295
MS-PCA 7153419-7153434
MS-PCA1242563– 1242566
MS-PCA1542047– 1542048
MS-PCA2405196–2405198
MS-PCA2409774- 2409775
MS-PCA2409907- 2409908
MS-PCA-IA-EC 034753-034894
MS-PCA-IA-EC 111852 -111854
MS-PCA-IA-EC 033066-033104
MSV 004301-0043021
MX 1142626-1142641
MX 1142724-1142731
MX 1263396-1263406
MX 1389851-1389852
MX 1394816-1394832
MX 1395980-1395982
MX 2045985-2045995
MX 2169954
MX 2175182-2175195
MX 2175196-2175211
MX 2217526-2217527
MX 2231010-2231014
MX 2261961-2261963
MX 2363703-236704
MX 2365819-2365823
MX 2661961-2661963

MX 3059957-3059980
MX 3129396-3129397
MX 3129496
MX 3171070-3171072
MX 3171073-3171074
MX 3171905-3171908
MX 3217128-3217131
MX 3251116-3251128
MX 3263480-3263481
MX 3264367-3264370
MX 4086171-4086179
MX 4129315-4129341
MX 5049798- 5049799
MX 5053800-5053801
MX 5066815-5066818
MX 5066863-5066864
MX 5066942-5066943
MX 5067022
MX 5095461-5095464
MX 5099114-5099116
MX 5103184-5103185
MX 5103233-5103235
MX 5117031-5117032
MX 5117033-5117034
MX 5121911-5121912
MX 6009686-6009689
MX 6025435
MX 6045525-6046634
MX 6047567-6047588
MX 6055840-6055844
MX 6097919-6097933
MX 6098017-6098022
MX 6109491-6109498
MX 6173120-6173130
MX 6184268-6184319
MX 7140791-7140792
MX 7141158
MX 7154940
MX 7155287-7155288
MX 7155600-7155605
MX 7157438-7157439
MX 7186069-7186070
MX 9025186-9025189
MX 9037644-9037645
MX 9037649-9037653
MX 9037656-9037660
MX 9037664-9037666
MX 9037667-9037669
MX 9037673-9037675
MX 9037682-9037684
MX 9038101-9038104
MX 9077917
NL2 0000071
NL2 0000094
NL2 0000175-0000176
NL2 0000177
NL2 0000182
NL2 0000578
NL2 0004273-0004274
NL2 0006918-0006919
NOV 00001330
NOV 00001412-00001413
NOV 00012429-00012431
NOV 0001378-0001379
NOV 00019339
NOV 00019380-00019382
NOV 00022240-00022241
NOV 00022242-00022262
NOV 00022294-00022307
NOV 00023675
NOV 00023718
NOV 00023783
NOV 00025907
NOV 00029182
NOV 00029183
NOV 00032647-00032648
NOV 00039774-00039777
NOV 00050916-00050917
NOV 00052662-00052664
NOV 00052665
NOV 00052711-00052713
NOV 00052786

NOV 00052833
NOV 00052834
NOV 00052842
NOV 00052863
NOV 00053523-00053524
NOV 00058370-00058394
NOV 00070388-00070419
NOV 00180461
NOV 00206606-00206614
NOV 00237966-00237984
NOV 00279350-00279386
NOV 00280175-00280176
NOV 00317758-00317771
NOV 00317786-00317799
NOV 00393046-00393062
NOV 00393274-00393307
NOV 00418040-00418055
NOV 00418617-00418619
NOV 00431215-00431278
NOV 00431599-00431624
NOV 00440032-00440043
NOV 00440044-00440046
NOV 00440185-00440186
NOV 00466362
NOV 00466756
NOV 00478302-00497366
NOV 00478359
NOV 00478382-00478390
NOV 00498169-00498187
NOV 00498188-00498204
NOV 00508054
NOV 00513028-00513033
NOV 00513036- 00513040
NOV 00513043-00513057
NOV 00513065-00513066
NOV 00513069
NOV 00513071-00513072
NOV 00513074-00513080
NOV 00513083
NOV 00516221
NOV 00516222-00516225
NOV 00516227
NOV 00516264-00516272
NOV 00516338
NOV 00516386
NOV 00517357-00517362
NOV 00517375-00517384
NOV 00517855-00517856
NOV 00518407-00518408
NOV 00520215
NOV 00526493-00526526
NOV 00528056-00528069
NOV 00531888
NOV 00532017-00532020
NOV 00533457-00533458
NOV 00538739-00538740
NOV 00540032-00540040
NOV 00540694
NOV 00540722-00540726
NOV 00541578
NOV 00541816
NOV 00541817
NOV 00542316-00542325
NOV 00543070-00543072
NOV 00544788-00544789
NOV 00544872-00544878
NOV 00545722-00545724
NOV 00546068
NOV 00570204-00570216
NOV 00570464-00570495
NOV 00571855-00571856
NOV 00616188
NOV 00616731-00616733
NOV 00617251-00617252
NOV 00617369-01491973
NOV 00618184
NOV 00630122-00630142
NOV 00631314-00631326
NOV 00637266-00637292
NOV 00664077-00664078
NOV 00642434-00642435
NOV 00642435-00642436

NOV 00649309-00649313
NOV 00649341
NOV 00649356
NOV 00649491-00649492
NOV 00649808
NOV 00649941
NOV 00650681-00650757
NOV 00654496-00654651
NOV 00660962-00661247
NOV 00660978-00660979
NOV 00661249-00661818
NOV 00661369-00661410
NOV 00661778
NOV 00662813
NOV 00665405
NOV 00671600
NOV 00671601
NOV 00671604-00671606
NOV 00671807
NOV 00672307
NOV 00672447
NOV 00673915-00673919
NOV 00674529
NOV 00680440-00680470
NOV 00684426
NOV 00684543
NOV 00685571-00685572
NOV 00686248
NOV 00686249
NOV 00686849
NOV 00686850-00686853
NOV 00686851-00686852
NOV 00686853
NOV 00686920
NOV 00687087-00687094
NOV 00687153-00687183
NOV 00687246-00687248
NOV 00687271-00687277
NOV 00687278
NOV 00687295-00687297
NOV 00687302
NOV 00687314-00687315
NOV 00690360-00690363
NOV 00690379-00690384
NOV 00692168-00692171
NOV 00700676-00700684
NOV 00700685-00700698
NOV 00701803-00701809
NOV 00704519-00704552
NOV 00704842
NOV 00705644-00705646
NOV 00707484-00707485
NOV 00707567-00707569
NOV 00707893
NOV 00707894
NOV 00709779-00709780
NOV 00709803-00709818
NOV 00709822-00709860
NOV 00709867-00709884
NOV 00710621
NOV 00713028-00713034
NOV 00713144-00713145
NOV 00713146-00713149
NOV 00713155-00713156
NOV 00713366
NOV 00713801-00713813
NOV 00713821-00713823
NOV 00720138
NOV 00721976-00721999
NOV 00725759-00725791
NOV 00727633-00727641
NOV 00728076-00728085
NOV 00728105
NOV 00734371-00734394
NOV 00749857-00749881
NOV00020493
NOV00032240
NOV00036684
NOV00044961
NOV00201293
NOV00237768-00237887
NOV00273397

NOV00417227-00417233
NOV00649917-00649918
NOV00686920-00686959
NOV00709822-00709860
NOV00720138-00720161
NOV00727633-00727641
NOV-23-002174-002210
NOV-25-000054-000056
NOV-25-000057-000061
NOV-25-000119-000120
NOV-25-000176
NOV-25-000336-000337
NOV-25-000401-000414
NOV-25-001156-001163
NOV-25-001643-001645
NOV-25-001655-001659
NOV-25-002046-002056
NOV-25-002491-002498
NOV-25-002528-002534
NOV-25-002718-002720
NOV-25-002738-002743
NOV-25-002765-002776
NOV-25-002791-002794
NOV-25-002941
NOV-25-002976-003006
NOV-25-006568-006578
NOV-25-008372-008374
NOV-25-023313
NOV-25-023426
NOV-25-026389-026390
NOV-25-026461-026468
NOV-25-026468
NOV-25-026752-026753
NOV-25-027385
NOV-25-027470-027473
NOV-25-027495-027497
NOV-25-027504-027505
NOV-25-028515
NOV-B00018964-00019043
NOV-B00029865-00029881
NOV-B00029916-00029918
NOV-B00029825-00029836
NOV-B00029858-00029863
NOV-B00029882-00029898
NOV-B00029902-00029904
NOV-B00029922-00029925
NOV-B00035809-00035814
NOV-B00046163
NOV-B00101839-00101841
NOV-B00104421-00102257
NOV-B00145327-00145345
NOV-B00150774-00150778
NOV-B00154327-00154338
NOV-B00154389-00154426
NOV-B00154613
NOV-B00154698
NOV-B00161001-00161016
NOV-B00169305
NOV-B00211452
NOV-B00314692-00314729
NOV-B00341309-00341334
NOV-B00365863-00365904
NOV-B00435798-00435811
NOV-B00435946-00435970
NOV-B00471686-00471698
NOV-B00516222-00516225
NOV-B00525074-00525084
NOV-B00542512-00542547
NOV-B00542515-00542547
NOV-B00544185-00544194
NOV-B00584305-00584312
NOV-B00602172-00602232
NOV-B00635696-00635736
NOV-B00635792-00635833
NOV-B00635837-00635845
NOV-B00635846-00635876
NOV-B00635919-00635943
NOV-B00635944-00635948
NOV-B00636106-00636116
NOV-B00636197-00636204
NOV-B00636237-00636262
NOV-B00636306-00636324

NOV-B00636362-00636381
NOV-B00636665-00636691
NOV-B00637139-00637172
NOV-B00637235-00637254
NOV-B00637266-00637292
NOV-B00642504-00642505
NOV-B00642585
NOV-B00642694-00642699
NOV-B00656734-00656748
NOV-B00656846
NOV-B00657540-00657550
NOV-B00661369-00661410
NOV-B00667073-00667119
NOV-B00668737
NOV-B00668860
NOV-B00673172-00673220
NOV-B00745195-00745221
NOV-B00745492-00745542
NOV-B00745579-00745600
NOV-B00745769-00745818
NOV-B00746832-00746833
NOV-B00749147-00749152
NOV-B00749829-00749856
NOV-B00749857-00749881
NOV-B00833306
NOV-B00833307-00833309
NOV-B00833346-00833349
NOV-B00833391-00833427
NOV-B00836853-00836862
NOV-B00840353-00840362
NOV-B00840356-00840362
NOV-B00853774
NOV-B00877582
NOV-B00877623-00877627
NOV-B00877697-00877701
NOV-B00932343-00932354
NOV-B00941714-00941723
NOV-B00941834-00941863
NOV-B00943082-00943096
NOV-B00950208-00950219
NOV-B00950715-00950722
NOV-B01024423-01024438
NOV-B01024443-00102469
NOV-B01024443-01024445
NOV-B01024446-01024469
NOV-B01024807-01024883
NOV-B01026193-01026198
NOV-B01026199-01026238
NOV-B01055255-01055263
NOV-B01056086-01056255
NOV-B01056256-01056487
NOV-B01056901-01057123
NOV-B01101261-01101282
NOV-B01105814-01105851
NOV-B01105944-01105947
NOV-B01110864-01101017
NOV-B01111492-01111613
NOV-B01112356-01112606
NOV-B01112607-01112824
NOV-B01113644-01113922
NOV-B01114345-01114641
NOV-B01152591-01152600
NOV-B01152642-01152645
NOV-B01192355-01192367
NOV-B01394553-01394559
NOV-B01395708-01395719
NOV-B01395720-01395732
NOV-B01396671-01396679
NOV-B01396680-01396689
NOV-B01396701-01396706
NOV-B01412084-01412108
NOV-B01413469-01413505
NOV-B01413613-01413615
NOV-B01413616-01413624
NOV-B01413629-01413648
NOV-B01420655-01402656
NOV-B01426057-01426062
NOV-B01426168-01426213
NOV-B01426508
NOV-B01426539-01426540
NOV-B01426652-01426679
NOV-B01426757-01426785

NOV-B01426796-01426870
NOV-B01433967-01433987
NOV-B01433988-01434007
NOV-B01434008-01434012
NOV-B01434013-01434020
NOV-B01434035-01434041
NOV-B01434126-01434156
NOV-B01434157-01434244
NOV-B01434462-01434479
NOV-B01434553-01434598
NOV-B01435409-01435433
NOV-B01440385-01440415
NOV-B01491183-01491213
NOV-B01491217-01491231
NOV-B01491440-01491455
NOV-B01491962-01491966
NOV-B01644564-01644570
NOV-B01644605-01644617
NOV-B01644687-01644689
NOV-B01645812-01645954
NOV-B01786593-01786609
NOV-B01857772-01857789
NOV-B01857797-01857802
NOV-B01857892-01857903
NOV-B01862772-01862778
NOV-B01869488-01869785
NOV-B01869786-01870036
NOV-B01870090-01870676
NOV-B01871520-01871784
NOV-B01872467-01872498
NOV-B01872562-01872660
NOV-B01873307-01873404
NOV-B01873705-01873900
NOV-B01886781-01886787
NOV-B01898883-01898963
NOV-B01904001-01904008
NOV-B01904035-01904037
NOV-B01904064-01904065
NOV-B01904075-01904076
NOV-B01904102-01904127
NOV-B01904373-01904422
NOV-B01905129-01905131
NOV-B01907105-01907259
NOV-B01917232-01917234
NOV-B01921246-01921254
NOV-B01921246-01921254
NOV-B01921266-01921273
NOV-B01924485-01924495
NOV-B01925892-01925899
NOV-B01929192-01929195
NOV-B01929533-01929538
NOV-B01929539-01929573
NOV-B01929641-01929645
NOV-B01932652-01935655
NOV-B01932672-01932687
NOV-B01932791-01932810
NOV-B01932840-01932853
NOV-B01939545- 01939590
NOV-B01939792-01939815
NOV-B01939861-01939865
NOV-B02011328-02011343
NOV-B02011392-02011406
NOV-B02020635-02020636
NOV-B02020635-02020636
NOV-B02418279-02418280
NOV-B02435208-02435218
NOV-B02435219-02435222
NOV-B02882173-02882196
NOV-B03687515-03687535
NOV-B03687517-03687535
NOV-B04016750-04016751
NOV-B04655478-04655479
NOV-B04799316
NOV-B04799317
NOV-B04799318-04799319
NOV-B04799320
NOV-B04799321-04799322
NOV-B04799323
NOV-B04799324
NOV-B04799325
NOV-B04799326-04799328
NOV-B04799329

NOV-B04799330
NOV-B04799331
NOV-B05424098-05424101
NOV-B05424510-05424544
NOV-B06507474-06507489
NOV-B06510118-06510143
NOV-B07466685-07466690
NOV-B07468996-07468998
NOV-B07563015-07563017
NOV-B07584816-07584817
NOV-B07585891-07585893
NOV-B07585894-07585895
NOV-B07730818-07731222
NOV-B07942554-07942555
NOV-B08136198-08136413
NOV-B13465526-13466527
NOV-B13528790-13528801
NOV-B13642621
NOV-B13642622
NOV-B13642900-13642908
NOV-B13643108-13643111
NOV-B13649415-13649416
NOV-B13649421
NOV-B13649422
NOV-B13649423
NOV-B13649424
NOV-B13649425
NOV-B13649426
NOV-B13649427
NOV-B13649431-13649441
NOV-B13651952-13651955
NOV-B13669521-13669533
NOV-B13861661-13861664
NOV-B13870330-13870334
NOV-B 14537458-14537460
NOV-B00018964-00019043
NOV-B00029865-00029881
NOV-B00029865-00029881
NOV-B00029902-00029904
NOV-B00029922-00029925
NOV-B00029922-00029925
NOV-B00035809-00035814
NOV-B00150774-00150778
NOV-B00154327-00154338
NOV-B00154389-00154426
NOV-B00161001-00161016
NOV-B0029825-0029836
NOV-B00365863-00365904
NOV-B00366268-00366272
NOV-B00366286-00366292
NOV-B00366346-00366352
NOV-B00366368-00366391
NOV-B00366392-00366403
NOV-B00435798-00435811
NOV-B00435946-00435970
NOV-B00471686-00471698
NOV-B00635696-00635736
NOV-B00635792-00635833
NOV-B00635837-00635845
NOV-B00635846-00635876
NOV-B00635916-00635943
NOV-B00635944-00635948
NOV-B00636237-00636262
NOV-B00636306-00636324
NOV-B00636362-00636381
NOV-B00745492-00745542
NOV-B00745579-00745600
NOV-B00746368-00746391
NOV-B00749829-00749856
NOV-B00749857-00749881
NOV-B00833391-00833427
NOV-B00853774
NOV-B00866160-00866163
NOV-B00877616-00877618
NOV-B00877623-00877627
NOV-B00882465-00882467
NOV-B00882468-00882471
NOV-B00941714-00941723
NOV-B00941834-00941863
NOV-B00943082-00943096
NOV-B00950208-00950219
NOV-B01024423-01024438

NOV-B01024443-01024445
NOV-B01026193-01026198
NOV-B01101179-01101202
NOV-B01101261-01101282
NOV-B01105814-01105851
NOV-B01105944-01105947
NOV-B01114525-01114527
NOV-B01152591-01152600
NOV-B01152642-01152645
NOV-B01154527-01154540
NOV-B01167360-01167387
NOV-B01396671-01396679
NOV-B01396680-01396689
NOV-B01396701-01396706
NOV-B01412084-01412108
NOV-B01413616-01413624
NOV-B01413629-01413648
NOV-B01425532-01425551
NOV-B01426051-01426062
NOV-B01426168-01426213
NOV-B01426539-01426540
NOV-B01426652-01426679
NOV-B01426757-01426785
NOV-B01433967-01433987
NOV-B01434008-01434012
NOV-B01434035-01434041
NOV-B01434126-01434156
NOV-B01434157-01434244
NOV-B01434480-01434491
NOV-B01434553-01434598
NOV-B01434661-01434694
NOV-B01435409-01435433
NOV-B01440385-01440415
NOV-B01450467-01450469
NOV-B01468558-01468564
NOV-B01491217-01491231
NOV-B01491962-01491966
NOV-B01644558-01644563
NOV-B01644564-01644570
NOV-B01644605-01644617
NOV-B01645812-01645954
NOV-B01786593-01786609
NOV-B01787803-01787825
NOV-B01787894-01787909
NOV-B01886753-01886770
NOV-B01886781-01886787
NOV-B01901855-01901866
NOV-B01901896-01901897
NOV-B01904013-01904021
NOV-B01904128-01904171
NOV-B01904884-01904892
NOV-B01905129-01905131
NOV-B01942855-01942863
NOV-B02002485-02002486
NOV-B02011328-02011347
NOV-B02011392-02011406
NOV-B02020632-02020636
NOV-B03687517-03687535
NOV-B03687635-03687657
NOV-B04491408-04491437
NOV-B04793934-04793955
NOV-B04799320
NOV-B04799321-04799322
NOV-B05017287-05017289
NOV-B05151790-05151837
NOV-B05398943-05398962
NOV-B07076008-07076010
NOV-B07327769
NOV-B07328689
NOV-B07328695-07328697
NOV-B07335240
NOV-B07335242
NOV-B07336038
NOV-B07336051-07336055
NOV-B07336056-07336058
NOV-B07336059-07336061
NOV-B07491747
NOV-B07542873-07542874
NOV-B07542875
NOV-B07562711
NOV-B07565714-07565715
NOV-B07585734-07585736

NOV-B07587390-07587394
NOV-B07604197-07605198
NOV-B07675897
NOV-B07701150-07701152
NOV-B08132601
NOV-B13465580-13465581
NOV-B13465582-13465583
NOV-B13528790-13528800
NOV-B13570730-13570745
NOV-B13649415-13649416
NOV-B13649421
NOV-B13649422
NOV-B13649423
NOV-B13649424
NOV-B13649427
NOV-B13649431-13649441
NOV-B13651952-13651955
NOV-B13652564-13652571
NOV-B13652900-13652908
NOV-B13653108-13653111
NOV-B13659425
NOV-B13659426
NOV-B13874944
NOV-B13874968
NOV-B14557034-14557057
NWA 000432-000435
NWA 000435
NWA 000197
NWP00002349-00002374
NWP00007760-00007783
NWP00008281-00008300
NWP00024863-00024864
NWP00024863-00024868
NWP00035565-00035569
SCM 010680-010683
WSUS 002495
WSUS 02168-02169
X 0600208
X 540709-540714
X 547090-547094
X 517411-517416

B.2 Depositions (Including Exhibits)

Adam Harral, December 12 and 13, 2001
Benjamin Slivka, February 5, 2009
Brad Chase, April 29, 2009
Brad Silverberg (Caldera), October 7, 1998
Brad Silverberg (Civil), June 22, 1994
Brad Silverberg, January 22, 2009
Brad Struss, January 14, 2009
Cameron Myhrvold, February 12, 2009
Charles Middleton, December 13, 2008
Chris Peters, February 24, 2009
Craig Bushman, November 18, 2008
Dale Christensen, February 5, 2009
Dana Charles Russell, February 6, 2009
David Acheson, November 19 and December 9, 2008
David Cole, February 26, 2009
David E. Miller (JCCP), November 2, 2001
David Hallmeyer, February 17, 2009
David LeFevre, January 7, 2009
Dean Clive Winn II, December 10, 2008
Donald Miller, March 4, 2009
Douglas Henrich, January 8, 2009
Eric Meyers (JCCP), September 28, 2001
Gary Lawrence Gibb, March 17, 2009
Gregory Richardson, December 13, 2001
James Lundberg, October 30, 2008
Jeff Raikes, January 27, 2009
Jim Allchin, (JCCP) April 10, 2002
Jim Allchin, January 8, 2009
John Ludwig, January 21, 2009
Jonathan Lazarus, January 27, 2009
Joseph Belfiore, January 13, 2009
Karl Ford, February 16, 2009
Kent Erickson, February 5, 2009
Mike Mathieu, January 13, 2009
Nathan Myhrvold, February 10, 2009
Noah Mendelsohn (JCCP) June 25, 2002
Nolan Larsen, November 17, 2008
Norman Thomas Creighton (JCCP) March 22, 2002
Norman Thomas Creighton, December 11, 2008
Paul Maritz (JCCP), October 24, 2001
Paul Maritz, January 9, 2009
Pete Higgins, December 17, 2008
Philippe Kahn, (JCCP) April 16, 2002
Richard Hume, March 24, 2009

Robert Frankenberg, March 25, 2009
Robert Kruger, January 29, 2009
Robert Muglia (JCCP), October 1, 2001
Robert Muglia, February 6, 2009
Robert Shurtleff, January 14, 2009
Russell Siegelman, March 5, 2009
Satoshi Nakajima, February 24, 2009
Scott Henson, December 17, 2008
Scott Kliger, February 25, 2009
Scott Raedeke, February 3, 2009
Stephen Madigan, January 15, 2009
Steven Sinofsky, December 18, 2008
Thomas Evslin, March 31, 2009
Tom Evslin, February 19, 2009
Willard E. Peterson, October 1, 2008
William Henry Gates (JCCP), February 28, 2002
William Henry Gates, March 4 and May 19, 2009

B.3 Other Materials

Novell Complaint, November 12, 2004

Technical Report of Ronald S. Alepin, August 29, 2002, and all documents and materials cited in that report

All Documents and Materials Specifically Cited in This Report (June 24, 2009)

Rebuttal Report of Ronald S. Alepin, November 4, 2002

Technical Report of Ronald S. Alepin, September 26, 2003

Expert Report of John K. Bennett, August 4, 2006, and all documents and materials cited in that report

Expert Report of David Martin, June 2, 2006

Expert Report of Ronald S. Alepin, June 2, 2006

Expert Report of John K. Bennett, July 14, 2003, and all documents and materials cited in that report

Technical Expert Report of David Martin, June 2, 2003

Microsoft and Novell Web Sites, and other freely available on-line materials

MAPI Cover Rev. 106 of MAPI 1.0, April 15, 1992

MAPI Rev. 106 of MAPI 1.0, April 26, 1992

Windows 7 Client Software Logo - Technical Requirements & Eligibility.xps

Microsoft Windows “Chicago” Reviewers Guide, Beta-1 (May1994)

Microsoft “Windows Logo Program - Requirements for the Windows Vista Logo Program for Software,” Version 1.0.0006, May 15, 2006

Microsoft “Windows Logo Program - Requirements for the Windows Vista Logo Program for Software,” Version 1.1.0001, October 30, 2006

Kotipalli, K., Support WebCasts Presentation: “Writing Shell Namespace Extension,” Microsoft Product Support Services (Undated)

Tom Haapanen, "Microsoft Windows Frequently Asked Questions," Copyrighted 1990-1994

"Chapter 4 Print Processors." Windows 95 DDK, March 1995

Chicago DDK CD, May 1994 SDK Quikview.doc

Chicago DDK CD, May 1994 SDK SHELLEXT.doc

Chicago DDK CD, October 1994 - Chapter 19: Electronic Mail and Information Exchange

Chicago DDK CD, October 1994 - Exchange Release Notes

Chicago DDK CD, October 1994 - SDK Quickview

Shephard, D., "What You Need to Do to Get the Windows 95 Logo When Windows 95 Ships," Microsoft Developer Network, February 1995

"Guide to the Windows 95 Logo Program," Microsoft, May 1995 (Updated)

"Technical Criteria - Windows 95 Logo Program," Microsoft Developer Network, Rev. February 3, 1995

"Windows 95 and NetWare: A Networking Insight" "Windows 95 Logo Update," January 1995: Q&A (February 1, 1995)

Letter to Developers for Logo Testing, February, 1995

Guide - New Key Support for Microsoft Windows Operating Systems and Applications (Optional Specifications)

Information about the Internet enabled file open dialog (Undated)

U.S. Patent Application Publication, *Dawson et al.,* Publication No. US 2007/00437001.A1, February 22, 2007

U.S. Patent Application Publication Nakajima et al. Patent No. 5,831,606 November 3, 1998

Shell extensions for an operating systems, U.S. Patent No. 6,008,806

Guide-File Objects and Shell Folders

Extending the Shell's Namespace

PerfectFit Print Services Press Release Insert (Undated)

WordPerfect Corporation, Answers.com

Chapter 4 - MAPI Architecture (Including Diagrams)

Chapter 1 – Print Spooler (same document as 14) Comparison Summary" Windows95 vs. Windows NT Workstation," Directions on Microsoft, January 1995

Microsoft's Information Exchange Strategy, Directions on Microsoft, v3n10

Microsoft's Desktop Operating System Strategy Part I, v2n12

Microsoft Window Messaging Subsystem: Transport Service Provider Interface, MAPI Version 1.0 – Draft Revision 106 for MAPI 1.0, April 15, 1992

Screenshot: Windows Chicago Desktops

Microsoft - Extending the Shell's Namespace

Shell Namespace Extensions Developer's Guide

FCEXT.H

GUIDE.HLP

Registry Extension Sample Application

shlobj_032996.h

shlobj_052695.h

shlobj_060994.h

shlobj_102894.h

MSJJUL96.EXE

Software\Chicoapp:
a) CHICOAPP.APS
b) CHICOAPP.C
c) CHICOAPP.EXE
d) CHICOAPP.H
e) CHICOAPP.RC

LISTVIEW.C (same as 94(m))

LISTVIEW.H (same as 94(n))

Software\enumdesk:
a) ABOUT.C
b) DISPATCH.C
c) DRAGDROP.CCP
d) DRAGDROP.H
e) ENUMDESK.APS
f) ENUMDESK.C
g) ENUMDESK.EXE
h) ENUMDESK.RC
i) GENERIC.C
j) GLOBALS.H
k) INI.C
l) INIT.C
m) LISTVIEW.C
n) LISTVIEW.H
o) MISC.C
p) PIDL.C
q) README.TXT
r) RESOURCE.H
s) TREVIEW.C (TREEVIEW.C)
t) TREVIEW.H (TREEVIEW.H)
u) WINMAIN.C

WordPerfect Developer's Toolkit

Chicago 05-94 Retail Extract

Chicago 10-94 Retail Extract

Windows 98 Extract

Chicoapp (same as 91)

Enumdesk (same as 94)

Internet Jumpstart Kit

June 1995 Win95 Beta Extract

May 1995 Win95 Beta (Spanish) Extract

Microsoft Windows 95 Internet Kit

Personal Operating Systems Windows News Win95 FTP

Windows95 FTP Site Archives

Undertaking given by IBM in 1984, Bulletin of the European Communities," Year 17, No.10, 1984

Tom Evslin, *How MAPI Beat VIM,* (available at http://blog.tomevslin.com/2006/01/index.html), Undated

Toronto Star: "Novell's WordPerfect takeover paying off", March 23, 1995

PC World: "PerfectOffice nearly lives up to its name", April 24, 1995

Findings of Fact ¶¶90-92

USA Today: "Novell Expansion Adds Skepticism," March 23, 1994

Asset Purchase Agreement, July 23, 1996

PC World: Reviews/Product Comparison, February 10, 1992

PC World: Word Processing Reviews, April 1992

Business Wire: "Novell to acquire WordPerfect Corporation and Purchase Borland's Quattro Pro spreadsheet business, March 21, 1994

Email: Reverse Engineering iShellFolder, August 30, 1995

Novell v. *Microsoft,* Plaintiff's Memorandum in Opposition to Defendant's Motion to Dismiss, February 22, 2005

Order, August 26, 2008

In Re Microsoft Corp. Antitrust Litigation, Novell's Objections and Responses for Production, March 2, 2009

GOinside: Athena: Goddess of War, Mail and News!, June 15, 1996

Microsoft Systems Journal: Extending the Windows Explorer with Name Space Extensions, July 1996

United States Patent: Shell Extensions for an Operating System, November 3, 1998

United States Patent: Shell Extensions for an Operating System, December 28, 1999

United States Patent: Method and System for Accessing Shell Folder Capabilities by an Application Program, March 19, 2002

Technical Report of Evan Ivie, August 26, 2002

Comes v. *Microsoft* Daily Trial Transcript, January 4, 2007

Comes v. *Microsoft* Daily Trial Transcript, January 5, 2007

Comes v. *Microsoft* Daily Trial Transcript, January 8, 2007

Comes v. *Microsoft* Daily Trial Transcript, January 9, 2007

Comes v. *Microsoft* Daily Trial Transcript, January 10, 2007

Comes v. *Microsoft* Daily Trial Transcript, January 11, 2007

Comes v. *Microsoft* Daily Trial Transcript, January 12, 2007

Investigative Report on Microsoft's Technical Exclusionary Actions, Undated

Extending Windows Explorer Namespaces with the .NET Framework, Undated

Shlobj 060994, Undated

Shlobj 102894, Undated