

Exhibit Q

04 - 1234

UNITED STATES COURT OF APPEALS FOR THE FEDERAL CIRCUIT

**EOLAS TECHNOLOGIES INCORPORATED and
THE REGENTS OF THE UNIVERSITY OF CALIFORNIA,**

Plaintiffs-Appellees,

v.

MICROSOFT CORPORATION,

Defendant-Appellant.

**APPEAL FROM THE UNITED STATES DISTRICT COURT FOR THE
NORTHERN DISTRICT OF ILLINOIS IN CASE NO. 99-CV-626,
JUDGE JAMES B. ZAGEL**

**BRIEF OF THE PLAINTIFFS-APPELLEES
EOLAS TECHNOLOGIES INCORPORATED and
THE REGENTS OF THE UNIVERSITY OF CALIFORNIA**

Martin R. Lueck
Jan M. Conlin
Richard M. Martinez
Munir R. Mehjee
ROBINS, KAPLAN, MILLER & CIRESI LLP
2800 LaSalle Plaza
800 LaSalle Avenue
Minneapolis, MN 55402-2015
(612) 349-8500

July 16, 2004

Attorneys for the Plaintiffs-Appellees

TABLE OF CONTENTS

	Page
CERTIFICATE OF INTEREST	i
STATEMENT OF RELATED CASES	xi
STATEMENT OF THE ISSUES	1
STATEMENT OF THE CASE AND THE FACTS	2
SUMMARY OF ARGUMENT	9
ARGUMENT	12
I. STANDARD OF REVIEW	12
II. THE COURT PROPERLY GRANTED JMOL ON VIOLA	12
A. “Viola” is missing elements of the ’906 claims	15
1. <u>DX37 does not anticipate</u>	17
a. “A computer program product for use in a system having at least one client workstation and one network server . . . wherein said network environment is a distributed hypermedia environment”	18
b. “type information . . . utilized by said browser to identify and locate an executable application”	25
c. “browser application to parse a first distributed hypermedia document”	26
2. <u>DX34 does not anticipate</u>	27

B.	DX34 and DX37 are not § 102(b) prior art because they were not in “public use”	28
C.	Both DX34 and the VOBJF architecture were abandoned, suppressed, and concealed under § 102(g)	32
D.	Microsoft did not present an obviousness case	35
III.	THE RULING ON INEQUITABLE CONDUCT WAS CORRECT	38
IV.	THE DISTRICT COURT COMMITTED NO ERROR OF CLAIM CONSTRUCTION	42
A.	Executable applications are not limited to standalone applications	42
B.	The browser’s use of outside resources is not in dispute	51
V.	MICROSOFT HAS LIABILITY FOR ITS EXPORT SALES UNDER § 271(F)	57
	CONCLUSION	63
	CERTIFICATE OF COMPLIANCE	64
	CERTIFICATE OF SERVICE	65

TABLE OF AUTHORITIES

	Page
CASES	
ACCO Brands, Inc. v. Micro Sec. Devices, Inc., 346 F.3d 1075 (Fed. Cir. 2003)	50
ACTV, Inc. v. Walt Disney Co., 346 F.3d 1082 (Fed. Cir. 2003)	57
Allied Colloids v. Am. Cyanamid Co., 64 F.3d 1570 (Fed. Cir. 1995)	30
Amgen Inc. v. Hoechst Marion Roussel, Inc., 314 F.3d 1313 (Fed. Cir. 2003)	52
Anchor Wall Sys., Inc. v. Rockwood Retaining Walls, Inc., 340 F.3d 1298 (Fed. Cir. 2003)	48
Apotex USA, Inc. v. Merck & Co., Inc., 254 F.3d 1031 (Fed. Cir. 2001)	13, 33
Applied Med. Res. Corp. v. U.S. Surgical Corp., 147 F.3d 1374 (Fed. Cir. 1998)	12, 15
Ashland Oil, Inc. v. Delta Resins & Refractories, Inc., 776 F.2d 281 (Fed. Cir. 1985)	37
AT&T Corp. v. Microsoft Corp., 2004 U.S. Dist. LEXIS 3340 (S.D.N.Y. 2004)	59
ATD Corp. v. Lydall, Inc., 159 F.3d 534 (Fed. Cir. 1998)	28, 37
Bayer AG v. Housey Pharms., Inc., 340 F.3d 1367 (Fed. Cir. 2003)	60

Beckman Instruments, Inc. v. LKB Produktur AB, 892 F.2d 1547 (Fed. Cir. 1989)	22
Brookhill-Wilk 1, LLC v. Intuitive Surgical, Inc., 334 F.3d 1294 (Fed. Cir. 2003)	44, 47
Cooper v. Goldfarb, 154 F.3d 1321 (Fed. Cir. 1998)	13, 17
Cybor Corp. v. FAS Techs., Inc., 138 F.3d 1448 (Fed. Cir. 1998)	12
Deere & Co. v. Int'l Harvester Co., 710 F.2d 1551 (Fed. Cir. 1983)	28
Delta-X Corp. v. Baker Hughes Prod. Tools, Inc., 984 F.2d 410 (Fed. Cir. 1993)	55
Diamond v. Diehr, 450 U.S. 175 (1981)	60
Dow Chem. Co. v. Astro-Valcour, Inc., 267 F.3d 1334 (Fed. Cir. 2001)	33
Ecolab Inc. v. Paraclipse, Inc., 285 F.3d 1362 (Fed. Cir. 2002)	54
Finnigan Corp. v. Int'l Trade Comm'n, 180 F.3d 1354 (Fed. Cir. 1999)	17, 24, 32
FMC Corp. v. Manitowoc Co., Inc., 835 F.2d 1411 (Fed. Cir. 1987)	38
Fujikawa v. Wattanasin, 93 F.3d 1559 (Fed. Cir. 1996)	33
Graham v. John Deere Co., 383 U.S. 1 (1966)	10, 14, 35, 37

Honeywell Inc. v. Victor Co. of Japan, Ltd., 298 F.3d 1317 (Fed. Cir. 2002)	47
Hybritech Inc. v. Monoclonal Antibodies, Inc., 802 F.2d 1367 (Fed. Cir. 1986)	13, 15
Imagexpo, L.L.C. v. Microsoft Corp., 299 F. Supp. 2d 550 (E.D. Va. 2003)	59
In re Alappat, 33 F.3d 1526 (Fed. Cir. 1994)	60, 61
Int'l Bus. Mach. v. United States, 201 F.3d 1367 (Fed. Cir. 2000)	58
Jamesbury Corp. v. Litton Indus. Prods., Inc., 756 F.2d 1556 (Fed. Cir. 1985)	25
Johns Hopkins Univ. v. CellPro, Inc., 152 F.3d 1342 (Fed. Cir. 1998)	61
Johnson Worldwide Assocs., Inc. v. Zebco Corp., 175 F.3d 985 (Fed. Cir. 1999)	42
Juicy Whip, Inc. v. Orange Bang, Inc., 292 F.3d 728 (Fed. Cir. 2002)	32
Karlin Tech. Inc. v. Surgical Dynamics, Inc., 177 F.3d 968 (Fed. Cir. 1999)	42, 47, 48
Kolmes v. World Fibers Corp., 107 F.3d 1534 (Fed. Cir. 1997)	12, 17, 35
Laporte v. Norfolk Dredging Co., 787 F.2d 1577 (Fed. Cir. 1986)	30
Liebel-Flarsheim Co. v. Medrad, Inc., 358 F.3d 898 (Fed. Cir. 2004)	46, 47

Modine Mfg. Co. v. U.S. Int’l Trade Comm’n, 75 F.3d 1545 (Fed. Cir.), cert. denied, 518 U.S. 1005 (1996)	46
Molins PLC v. Textron, Inc., 48 F.3d 1172 (Fed. Cir. 1995)	38
Monon Corp. v. Stoughton Trailers, Inc., 239 F.3d 1253 (Fed. Cir. 2001)	41
Motorola, Inc. v. Interdigital Tech. Corp., 121 F.3d 1461 (Fed. Cir. 1997)	25
N. Telecom Ltd. v. Samsung Elecs. Co., Ltd., 215 F.3d 1281(Fed. Cir. 2000)	48, 52, 53
Netscape Communications Corp. v. Konrad, 295 F.3d 1315 (Fed. Cir. 2002)	30
Norian Corp. v. Stryker Corp., 363 F.3d 1321 (Fed. Cir. 2004)	38
NTP, Inc. v. Research in Motion, Ltd., 261 F. Supp. 2d 423 (E.D. Va. 2002)	59
Nystrom v. Trex Co., Inc., 2004 U.S. App. LEXIS 13407 (Fed. Cir. 2004)	44
Panduit Corp. v. Dennison Mfg. Co., 810 F.2d 1561 (Fed. Cir. 1987)	14, 37
Pellegrini v. Analog Devices, Inc., 2004 U.S. App. LEXIS 14017 (Fed. Cir. 2004)	58
Sandt Tech., Ltd. v. Resco Metal & Plastics Corp., 264 F.3d 1344 (Fed. Cir. 2001)	24, 34
Scripps Clinic & Research Found. v. Genentech, Inc., 927 F.2d 1565 (Fed. Cir. 1991)	16, 17, 36

Southwest Software, Inc. v. Harlequin Inc., 226 F.3d 1280 (Fed. Cir. 2000)	59
Teleflex, Inc. v. Ficosa N. Am. Corp., 299 F.3d 1313 (Fed. Cir. 2002)	16, 54
TI Group Auto. Sys., Inc. v. VDO N. Am., L.L.C., 2004 U.S. App. LEXIS 13445 (Fed. Cir. 2004)	44
Union Carbide Chems. & Plastics Tech. Corp. v. Shell Oil Co., 308 F.3d 1167 (Fed. Cir. 2002)	13
Viskase Corp. v. Am. Nat'l Can Co., 261 F.3d 1316 (Fed. Cir. 2001)	18
Vitronics Corp. v. Conceptronic, Inc., 90 F.3d 1576 (Fed. Cir. 1996)	52
W.L. Gore & Assocs., Inc. v. Garlock, Inc., 721 F.2d 1540 (Fed. Cir.1983)	25, 29, 30
WMS Gaming, Inc. v. Int'l Game Tech., 184 F.3d 1339 (Fed. Cir. 1999)	61
Woodland Trust v. Flowertree Nursery, Inc., 148 F.3d 1368 (Fed. Cir. 1998)	13, 30

STATUTES

35 U.S.C. § 102	22
35 U.S.C. § 102(b)	9, 10, 12, 14, 15, 27, 28, 30, 32
35 U.S.C. § 102(g)	9, 10, 12-15, 27, 32, 35
35 U.S.C. § 102(g)(2)	13
35 U.S.C. § 103	9, 12-15

35 U.S.C. § 271(a)	62
35 U.S.C. § 271(f)	1, 11, 12, 57-62
35 U.S.C. § 271(f)(1)	58
35 U.S.C. § 271(f)(2)	58
35 U.S.C. § 271(g)	60
35 U.S.C. § 282	14
F.R.E. 402	14, 17, 34
F.R.E. 403	14, 17, 34
Fed. R. Civ. P. 61	28

OTHER AUTHORITIES

IBM Dictionary of Computing (10th ed. 1994)	45
Manual of Patent Examining Procedure § 2106 (8th ed. 2003)	59, 61
Manual of Patent Examining Procedure § 2106.01(8th ed. 2003)	59
Manual of Patent Examining Procedure § 2106.02 (8th ed. 2003)	59
Microsoft Press Computer Dictionary (2d ed. 1994)	43, 45
Microsoft Press Computer Dictionary (3d ed. 1997)	44
The American Heritage Dictionary (3d ed. 1992)	59, 61
Webster’s Third New International Dictionary (3d ed. 1986)	59, 61

STATEMENT OF RELATED CASES

No appeal in or from the same civil action was previously before this Court or any other appellate court.

The reexamination proceedings of U.S. Patent No. 5,838,906 referenced by Microsoft was initiated by the PTO Director after a concerted effort by Microsoft. A003190-203. The art on which Microsoft urged and the Director initiated reexamination is entirely unrelated to the Viola software cited by Microsoft here. Id.; A002529-37.

The pending appeals in NTP, Inc. v. Research In Motion, Ltd., Case No. 03-1615, and AT&T Corp. v. Microsoft Corp., Case No. 04-1285, each raise issues concerning 35 U.S.C. § 271(f).

Microsoft attempted to reissue U.S. Patent No. 5,801,701 to claim the claims of U.S. Patent No. 5,838,906. Application Number 09/442,070. That attempt failed before the PTO, and an appeal is pending.

Counsel is unaware of any other case, in this or any other court, known to be directly affected by this Court's decision in this appeal.

STATEMENT OF THE ISSUES

1. Whether the District Court properly granted JMOL on Microsoft's anticipation and obviousness defenses based upon the offered Viola software.
2. Whether the District Court properly rejected Microsoft's inequitable conduct defense, finding that Dr. Doyle made no intentional omission of material information to the PTO.
3. Whether the District Court correctly construed the term "executable application."
4. Whether the District Court correctly construed the term "utilized by said browser to identify and locate" and gave permissible jury instructions relating to that construction.
5. Whether the District Court correctly held that Microsoft could not avoid liability under 35 U.S.C. § 271(f) for placing its final software code on a "golden master" in the United States and then exporting it abroad to be installed.

STATEMENT OF THE CASE AND THE FACTS

The ability to bring rich interactive content to a user's computer screen by merely typing a web address into an Internet browser has transformed the way information is acquired and delivered throughout the world. The invention at issue in this case, United States Patent No. 5,838,906, took the World Wide Web from a static environment, in which a user could view text or images, to today's seamlessly interactive environment, in which millions of web users routinely interact with web pages.

In 1993, while web developers were struggling with embedding static images, researchers at the University of California's Innovative Software Systems Group were already developing the potential for the web to become a platform for embedding interactive applications. Among their projects was the development of an on-line medical journal, which would make available the latest medical developments to physicians around the world. A100619-27. As part of this work, the '906 inventors struggled to build a database describing the human embryo to assist scientific and medical research. Id. The inventors had to find a way to deliver an enormous, often changing, data set with an associated application to research personnel who worked on 1993 era computers with limited processing power. Id.

The invention of the '906 patent is the result of that effort. From the standpoint of the user, the '906 invention enables the use of embedded applications within a web page by doing no more than typing the uniform resource locator ("URL") of the web page into the browser. A100597. Rather than adding the voluminous code for each potential executable application to the browser, the '906 invention gives the browser the ability to identify and locate the necessary application, seamlessly integrate it into the web page, and to deliver it automatically to the user's screen. A100598; A100622-27; A100746-47. Although Microsoft criticizes the invention as "adding just 305 lines" of code to a browser (MS Br. at 9), this economical use of code is part of the elegant solution to interactivity brought to the World Wide Web by the '906 inventors.

When it was announced, the invention received extensive publicity in the United States and abroad, including prominent articles in newspapers and periodicals. A100703. Today, this powerful tool is widely used to deliver interactive content such as stock and financial information, scientific data, games, product information and interactive maps. A100852-55; A100919-20.

Claims 6 (apparatus) and 1 (method) of the '906 patent are at issue. The claims require a browser with certain properties. The browser must operate in a "distributed hypermedia environment." A150020. The web browser locates the web page

(“distributed hypermedia document”) using a URL. A150012. The browser parses the text of the web page, including an “embed text format” that specifies “at least a portion of an object external to the” web page (like a visual object requiring large volumes of data) that has “type information associated with it,” – e.g., what type of application to run (spreadsheets, database, etc.). A150020. The type information is then utilized by the browser to identify and locate an executable application, which is automatically invoked to enable interactive viewing of the object. Id.

Microsoft adopted the '906 technology in 1996 with its Internet Explorer 3.0 and has included the technology in every copy of Windows sold since. A100824-26; A100912-13. Microsoft introduced “Active X controls,” its name for its executable applications that run with its '906-enabled browser, in response to competitive threats from Netscape’s Navigator browser. Microsoft feared that Netscape’s ability to deliver applications to the user’s screen within a web page over any underlying operating system (as opposed to only Microsoft’s proprietary Windows operating system) would “commoditize” Windows and threaten the enormous profits Windows commanded. A100904-25; A101504; A151485-87; A151523; A151529; A151534; A151537; A151540; A151580-84; A151589-90; A151596-97; A151604; A151607; A151614-15; A151618; A151621; A151625; A151639.

Microsoft sought to invalidate the '906 patent claims through blending together an amalgamation of references regarding software known as "Viola" developed by Pei Wei. The only software that Microsoft offered as the browser required by the '906 patent claims were two versions of the Viola software, DX34 (May 12, 1993) and DX37 (May 27, 1993). No other software was offered, and no printed publication revealed whether any other Viola browser met all elements of the '906 claims.

Both DX34 and DX37 employed a <VOBJF> tag to achieve the functionality that Microsoft has relied on to try to meet all of the limitations of the '906 claims. A101128; A101172. That architecture allowed – *when all of the parts were on a single machine* – the loading of a "page" called Testplot.htmml. The "page" contained a <VOBJF> tag that contained the precise file path of an "object" (Plot.v) on the same machine, which in turn contained the precise file path of an "executable application" (Vplot) on the same machine. A100844-47; A101206; A101308-09.¹

Ultimately, the evidence indisputably showed that *neither* version of Viola actually operated or was capable of use, as required by claims 1 and 6, in a *distributed*

¹ Both DX34 and DX37 had this architecture, and Microsoft represented that there was no material difference between them. A101127. In fact, setting aside other issues with the code addressed elsewhere, only DX34 had the functionality that supposedly allowed the browser to load an object retrieved from a different machine than the one on which the browser was located. A101161-64; A101173-75; A101203-05.

hypermedia network environment – one where the browser is on one computer and the web page on another (A101308). See pages 18-25, 27-28, infra.

Initially, Microsoft asserted otherwise, representing that the Viola code “is as hard evidence as one could ever get. We have actual code. We can tell what the code is capable of. We can tell what it runs by running it today.” A101128. As the district court pressed Microsoft for more details regarding Viola, Microsoft’s counsel stated that, to get the proffered Viola browsers to work in a distributed hypermedia network environment: “[W]e don’t have to make any fixes or anything like that to the Viola code or to the server code.” A101130.

But, when undertaking its demonstration of this assertion, Microsoft actually had to create a new server system that existed neither in 1993 nor anytime else to make up for the deficiencies in the Viola code. Wei admitted to these changes after he attempted to demonstrate DX34 and DX37:

Q. So the software that your testplot.HMML page was on when you downloaded it in the demonstration you showed the Court today was not in existence at the time you made the demonstration of the May 12, 1993 demonstration to Sun, correct sir?

A. That’s correct.

Q. All right. Now, the source code, as you know, was modified, right?

A. Yes.

Q. What you talked about in terms of changing the May 12, 1993, Viola code with reference to the 0.9 and 1.0 HTTP standard, you said that that change could have been done in the browser code, correct?

A. In the World Wide Web library code.

Q. Well, in the -- all right. But in the Viola code, correct?

A. It's -- this is not a Viola code per se. It's part of the system.

Q. Let me change my question then.

The change could have been made at the Viola end of the system instead of the server end of the system, right?

A. Yes, yes you could do that too.

Q. But it was chosen that the changes would be made in the server side of the system and represented to the court that is worked without any changes to the Viola side of the system; isn't that right?

A. That's right.

A101177; A101186.

The patent limitation of use in a distributed hypermedia network environment is not the only limitation Viola did not meet. As discussed below, the two Viola browsers failed to meet two other limitations of claims 1 and 6: The requirement that the browser parse a web page; and the requirement that the browser, not the object,

identify and locate the executable application (for interaction with the object). See pages 25-28, infra.

The facts proven at trial also showed that neither the two versions of Viola nor the VOBJF architecture was public. Wei demonstrated the May 12, 1993 version of Viola (DX34) to two Sun Microsystems engineers, but when he did so, *he never demonstrated the functionality that was unique to DX34*. A101176-77. Nor did he ever give the code of DX34 to Sun for Sun to discover the functionality itself. A101174. In fact, DX34 was *never* given to anyone. A101175.

Wei sent the later version, DX37, to the Sun engineers after the demonstration. They were unable, however, to make it work at all. A101207; A170030. And even when sending the Sun engineers the DX37 code, he ensured that only they, and not others, would receive it. A101207; A151665; A170024.

Beginning in the fall of 1993, Wei distributed some version of Viola to three individuals, but Microsoft did not introduce that version (or any beyond DX34 and DX37) or any printed publication that documents its functionality, so the content or capability of that version is unproved. Although Microsoft characterizes this as an “alpha” release, there was no general “alpha” release of Viola. A101207-08; A151531. The distribution method, moreover, allowed only the recipient to receive

the code, and made it “nearly impossible” for the public to “get their hands on it.”
A101207-08; A101211; A151532; A170035.

Ultimately, by January 17, 1994, Wei abandoned his entire VOBJS architecture. A101172-73; A101175; A101184. Microsoft’s so-called “beta” release in February 1994 incorporated a new architecture based on the implementation of a <LINK> tag (developed after the ’906 invention) with different capabilities, not the VOBJS architecture. Id. Microsoft did not meet the standards for proving invalidity through this architecture – no code was ever produced, let alone proffered or demonstrated. No prior art was proffered that revealed the state of development of VOBJS architecture by the time of this release, or that corroborated Wei’s claims that the new architecture met the elements of the ’906 invention.

SUMMARY OF ARGUMENT

Microsoft is not entitled to a new trial on its invalidity defense – public use under § 102(b), prior invention under § 102(g), and obviousness under § 103. For overlapping and multiple reasons, the district court correctly excluded DX34 (offered by Microsoft only for § 102(b) and § 102(g)) and correctly granted JMOL after Microsoft’s attempt to prove invalidity with DX37.

First, no corroboration exists of any browser – needed for public use under § 102(b) and reduction to practice under § 102(g) – except DX34 and DX37. But

neither DX34 nor DX37 contains all elements of claim 1 or claim 6 of the '906 patent, as required for anticipation under both § 102(b) and § 102(g). As a matter of law, neither Viola browser anticipates.

Second, as to § 102(b), Microsoft's invalidity defense fails for the additional reason that neither DX34 nor DX37 was in public use. The functionality that was unique to DX34 was not demonstrated or shown to the Sun engineers or any other member of the public. Moreover, DX37 was never successfully used by Sun and was not distributed to the public. Wei's asserted subsequent distributions made it "nearly impossible" for the public to get the code.

Third, as to § 102(g), Microsoft's defense fails on grounds of abandonment, suppression, and concealment. Wei changed the functionality that was unique to DX34, now touted by Microsoft as key, and it did not appear in any subsequent version of Viola. Thus, it was abandoned by Wei between the May 12, 1993 DX34 and the May 27, 1993 DX37. Ultimately, the entire VOBJF architecture, by Wei's own admission, was abandoned as of January 1994.

Fourth, Microsoft utterly failed to prove obviousness under Graham v. John Deere Co., 383 U.S. 1 (1966). Despite assuring the court that it would present an obviousness defense, Microsoft reversed course and presented expert testimony that

the '906 patent was invalid exclusively because of anticipation. Microsoft did not attempt to resuscitate obviousness until the evidence was already closed.

As to Microsoft's inequitable conduct defense, the district court correctly held that there was no inequitable conduct in inventor Doyle's alleged failure to disclose what he had heard about Viola. What information Doyle had was not material. The court's factual and credibility determinations in finding that the applicants did not believe that Viola was material, and had no intent to mislead the PTO, were supported by the record.

The district court committed no error respecting claim construction. As to "executable application," the court properly refused to import "standalone" into the construction because the ordinary meaning and intrinsic evidence do not support that restriction.

The district court also correctly construed the claim element requiring the browser to utilize type information to identify and locate the executable application. Microsoft acknowledged that the operating system can be involved in this task, and itself proposed the language adopted in the court's construction and jury instructions.

Finally, the district court properly held that Microsoft faced liability for its export sales under § 271(f) because the Windows code on the golden master created in the United States is a "component" of the patented computer program product

invention. When the code is installed abroad, there has been a “combination” that would infringe if it occurred in the United States. Section 271(f) thus applies.

ARGUMENT

I. STANDARD OF REVIEW

While Microsoft is correct that “de novo” review applies to many of the issues raised here, “[s]imply because a particular issue is denominated a question of law does not mean that the reviewing court will attach no weight to the conclusion reached by the tribunal it reviews.” Cybor Corp. v. FAS Techs., Inc., 138 F.3d 1448, 1463 (Fed. Cir. 1998) (en banc) (Bryson, J., concurring).

In addition to the standards of review set forth by Microsoft, the district court’s evidentiary rulings are reviewed under an abuse of discretion standard: To obtain a new trial, Microsoft must show that the court abused its discretion in excluding the evidence, and “such ruling prejudiced its substantial rights and was thus not harmless error.” Kolmes v. World Fibers Corp., 107 F.3d 1534, 1542 (Fed. Cir. 1997).

II. THE COURT PROPERLY GRANTED JMOL ON VIOLA

Microsoft asserts three grounds of invalidity – § 102(b), § 102(g), and § 103 – based on Wei’s software under the name Viola. Section 102(b)’s public use defense contains two relevant requirements. Microsoft had to show an actual version of Viola that (1) contains each and every element of the claimed invention, (see Applied Med.

Res. Corp. v. U.S. Surgical Corp., 147 F.3d 1374, 1378 (Fed. Cir. 1998)), and (2) was in *public* use so as to make the invention “available to the public.” Woodland Trust v. Flowertree Nursery, Inc., 148 F.3d 1368, 1371 (Fed. Cir. 1998).

Section 102(g) likewise requires that Microsoft have proved some actual version of Viola that contains every element of the claimed invention. Hybritech Inc. v. Monoclonal Antibodies, Inc., 802 F.2d 1367, 1379 (Fed. Cir. 1986). To meet the invention and reduction-to-practice requirements of § 102(g), Microsoft had to show that Wei (1) constructed an embodiment that performed all the limitations of the claim, and (2) determined that the invention would work for its intended purpose. Union Carbide Chems. & Plastics Tech. Corp. v. Shell Oil Co., 308 F.3d 1167, 1189 (Fed. Cir. 2002); Cooper v. Goldfarb, 154 F.3d 1321, 1327 (Fed. Cir. 1998). Still further, the asserted Wei invention must have entered the public domain rather than been “abandoned, suppressed, or concealed.” 35 U.S.C. § 102(g)(2); Apotex USA, Inc. v. Merck & Co., Inc., 254 F.3d 1031, 1035 (Fed. Cir. 2001).

For its § 103 obviousness defense, Microsoft had, at a minimum, to show facts sufficient to establish by clear and convincing evidence (1) the scope and content of the prior art, (2) the differences between the prior art and the claims at issue, (3) the level of ordinary skill in the art, and (4) the objective evidence of nonobviousness.

Graham, 383 U.S. at 17-18; Panduit Corp. v. Dennison Mfg. Co., 810 F.2d 1561, 1566 (Fed. Cir. 1987).

The district court properly disposed of each invalidity claim under these standards. When trial began, Microsoft's Viola invalidity case rested on DX37 – it was the only version of Viola disclosed in its Notice Pursuant to 35 U.S.C. § 282. A003521-29. After Plaintiffs-Appellees' expert Dr. Felten showed the jury why DX37 did not meet the '906 claim elements, A100844-47, Microsoft changed course and tried to fill the gaps in DX37 with DX34 and a host of emails. A101089-90. The court then conducted a lengthy voir dire of Wei as an offer of proof regarding all of the Viola "evidence," heard argument, examined Wei's deposition, and examined portions of the report and deposition of Microsoft's expert Dr. Kelly. A101126-34; A101153-96.

After the voir dire, the district court excluded DX34 as evidence of the claimed Wei invention. It concluded that if DX34 embodied the invention, it was abandoned, suppressed, and concealed as a matter of law under § 102(g), and not in public use under § 102(b). A101186-93. The court excluded the Viola emails under F.R.E. 402 and 403. Id. Microsoft asserted no other grounds – including § 103 – for the admission of DX34. A101089-90; A101186-96.

Microsoft proceeded with DX37. At the close of evidence, the district court granted Plaintiffs-Appellees' JMOL on invalidity. It concluded that DX37 did not anticipate because it did not meet all of the elements of the '906 claims and that Microsoft did not present evidence required to show obviousness. A101362-63.

Both the exclusion and JMOL rulings were correct. First, DX37 cannot meet the common requirement of § 102(b) and § 102(g) anticipation – that it was a browser meeting every element of the '906 claims. In fact, DX34 also is missing the same elements, because it shares the same VOBJF architecture – an alternative grounds to affirm the district court's exclusion. Second, neither DX34 nor DX37 (given Wei's conduct with these browsers and Wei's ultimate abandonment of the VOBJF architecture) can meet the requirements of § 102(b) – public use – and § 102(g) – lack of abandonment, suppression, or concealment. Finally, Microsoft utterly failed to prove the requirements of § 103.

A. “Viola” is missing elements of the '906 claims

To show anticipation by Viola for either its § 102(b) or § 102(g) defenses, Microsoft needed, at a minimum, clear and convincing proof that Wei had an actual, reduced-to-practice browser that itself met *every limitation* of the '906 claims. Applied Med. Res. Corp., 147 F.3d at 1378; Hybritech Inc., 802 F.2d at 1379. A finding of anticipation is “not supportable if it is necessary to prove facts beyond

those disclosed in the reference in order to meet claim limitations”; extrinsic evidence may be used to understand the content of asserted prior art but “*not to fill in gaps in the reference.*” Scripps Clinic & Research Found. v. Genentech, Inc., 927 F.2d 1565, 1576 (Fed. Cir. 1991) (emphasis added). See Teleflex, Inc. v. Ficosa N. Am. Corp., 299 F.3d 1313, 1335 (Fed. Cir. 2002) (rejecting attempt to combine references for anticipation).

The only browsers that Microsoft offered as meeting all the limitations of the '906 patent claims – DX34 and DX37 – do not do so: They are missing important claim elements. Microsoft impermissibly tried to fill the gaps in these references through mere testimony regarding other supposed versions of Viola. No code or printed publication was offered to corroborate that any other version of Viola met all the claim elements. Since no other actual browser, or even documentation proving the elements of any such browser, was offered by Microsoft, it simply failed to prove that there was any other Viola browser that anticipated the '906 invention.

Contrary to Microsoft's suggestion, it is legally immaterial whether the only code Plaintiffs-Appellees provided *for the '906 invention* was contemporaneous with their patent application. MS Br. at 22 n.10, 38. It is undisputed that the '906 inventors reduced their invention to practice with each element of the claimed invention, as demonstrated by that code, by the application, and by the issued patent.

The question with Wei is not whether he reduced some generic Viola browser to practice, but whether what he specifically reduced to practice had each element of the '906 claims. There is no Viola code whatsoever that contains each element of the '906 invention; there is nothing beyond Wei's bald – and legally insufficient – after-the-fact assertions that he ever had any such code.

In short, given the bedrock requirements of corroboration and reduction to practice, see Finnigan Corp. v. Int'l Trade Comm'n, 180 F.3d 1354, 1367-69 (Fed. Cir. 1999); Cooper, 154 F.3d at 1328, only DX34 and DX37 were even candidates for the allegedly invalidating browser, and they fail.²

1. DX37 does not anticipate

When forced during trial to state its position, Microsoft's counsel *admitted* that it needed to fill gaps in DX37 to make its anticipation case:

THE COURT: So basically your answer to Scripps is, yes, there's a gap here, and, yes, we're filling this gap.

MR. BAUMGARTNER: Correct.

² Microsoft points to an email by Wei about reduction to practice in some supposed earlier Viola browser. MS Br. at 5, 31. But this email plainly does not corroborate satisfaction of all elements of the '906 claims. A101172; A101186-87; A170005-06. Nor does the other Viola "evidence" cited by Microsoft corroborate any later reduction to practice of a browser containing the VOBJF functionality, let alone one meeting all of the '906 claim elements. A170003-7; A170037-43; A170046-77. The district court excluded this evidence under F.R.E. 402 and 403. A100457-59; A100466-73; A101186-93; A102019-26. These rulings were not an abuse of discretion. See Kolmes, 107 F.3d at 1542.

A101189. That admission is binding. And, in any event, Microsoft’s counsel was correct. There was no evidence from which a jury could find the following elements in the DX37 Viola browser – that (1) it was a computer program product for use in a distributed hypermedia network environment; (2) the “browser” utilized “type information” to “identify and locate” an executable application; and (3) the “browser” “parse[d] a first distributed hypermedia document.”³

- a. *“A computer program product for use in a system having at least one client workstation and one network server . . . wherein said network environment is a distributed hypermedia environment”*

A *distributed* hypermedia network environment is one in which the browser is on one computer, and the hypermedia document is on another. A101308.⁴

There is no corroborated evidence that DX37 was *actually used* in such an environment. A101203-05. When Wei tried to recant his previous sworn testimony that he could not recall ever having done so, he admitted that he had no “evidence” of any such use. A101203.

³ Microsoft claims that the district court never specifically identified what elements were missing from Viola. In fact, the record was clear on the missing elements, both through trial testimony and arguments of counsel. *E.g.*, A101126-34; A101154-92; A101347; A101352. This is sufficient to uphold a district court’s grant of JMOL. *Viskase Corp. v. Am. Nat’l Can Co.*, 261 F.3d 1316, 1327 (Fed. Cir. 2001).

⁴ Microsoft assiduously avoids the term “*distributed*” in its arguments regarding what Viola is and what it supposedly “taught.” MS Br. at 5, 6, 34, 43.

In fact, Microsoft failed to prove that, *as written*, DX37 was *capable of use* in such an environment. Microsoft tried to get Viola to work in a distributed hypermedia network environment after much effort by making fundamental changes to the Viola code and by creating a unique server – changes necessary to even allow Microsoft to try to make Viola capable of such use. A101160-61; A101173-78; A101203-05. Even with the efforts of the world’s largest software company, DX37 still would not work over the Internet, and Wei did not know why. *Id.* Now, Microsoft seeks to reverse the district court’s ruling with only the naked testimony of its expert that Viola might yet be made to work in a distributed hypermedia environment if some other changes were made – though Microsoft does not specify how they could be accomplished. MS Br. at 45. This does not meet Microsoft’s burden.

The fundamental structure of VOBJF architecture (in both the DX34 and DX37 versions, in fact) explains why Microsoft could not show that it could be used in a distributed hypermedia network environment. The VOBJF architecture was geared to use on a single machine: As designed, the Viola.www browser loaded a “page” called Testplot.hmml on the same machine as the browser. The Testplot.hmml page contained a <VOBJF> tag that contained the precise file path to the “object” (Plot.v), which was located on that same machine. The browser loaded the “object” Plot.v,

a Viola script or program, from that precise file path. That “object” contained the precise file path to what Microsoft called an “executable application” (Vplot), also on the same machine. The <VOBJF> tag in the “hypermedia document” must be recoded for each computer on which the browser was used to specify the particular file path for the object on that machine. A100844-47; A101164; A101206; A101308-09.

In order to begin to make Viola capable for use in a *distributed* hypermedia network environment, Microsoft had to change this structure. First, Microsoft had to move both the Testplot.hmml “page” and the plot.v “object” to a server. Then, it had to place plot.v in the file structure on that same server as specified in the recoded <VOBJF> tag. A101176-77; A101202-03.

Second, because DX37 was not written in a way that would work with any server ever in existence, Microsoft created an entirely new server protocol for the demonstration that it claimed would prove that Viola could be used in a distributed hypermedia network environment. A101176-78; A101186; A101203. Microsoft altered post-1993 server code on the server on which it placed the hypermedia document. Id. The alterations created a brand new amalgamation of the 0.9 HTTP protocol and 1.0 HTTP protocol, one never in existence before Microsoft created it in 2003. Id. While Microsoft claims that these modifications made by its consultants

(not disclosed until cross-examination ferreted them out) simply made Viola compatible with the 0.9 HTTP protocol in existence in May 1993, MS Br. at 21 n.9, Wei admitted that his Viola code would not work with the 0.9 protocol. A101178; A101203. Nor, as Wei admitted, could Viola work with the 1.0 protocol. A101203. The reason, discovered only after the extensive effort of Plaintiffs-Appellees in pouring over the voluminous code, is that Viola sends requests to a server using one protocol, but accepts documents back only if sent under another protocol. A101177-78; A101203.⁵

Third, in addition to this server issue, DX37 required additional changes to the Viola files that implemented the <VOBJF> tag, changes that Wei claimed would allow it to load an object if one could have been retrieved over the Internet. A101161-64; A101173-75; A101203-05. In Microsoft's attempt to demonstrate DX37 in Wei's voir dire, Wei, while in the courtroom, made changes to these Viola files in an attempt to make it work. A101164. Microsoft's demonstration still failed. Id. Wei testified that he did not know why the Viola code did not work, and he conceded that there may be other problems with the code of which he was unaware

⁵ Thus, Microsoft's statement that DX37 will work as long as "the server sends documents using HTTP version 0.9" is misleading by omitting half of the story. MS Br. at 45. There is no evidence that a server ever existed that would accept requests from a browser under the 1.0 protocol, and then send documents using the 0.9 protocol, except the one Microsoft created.

that would still prevent it from loading an object retrieved in a distributed hypermedia network environment. A101173-75; A101203-05.

Microsoft attempts to dismiss its failure of proof in two ways. First, it argues that these fundamental problems with Viola are mere “bugs” and thus DX37 still “taught” all elements of the ’906 claims. MS Br. at 43-46. In the district court, however, Microsoft acknowledged that DX37 did not meet the § 102 standard that all claim elements are inherent or necessary in the reference, and that it was left only with an obviousness argument. A101189.

DX37, a piece of software, teaches only the functions its code performs; it does not “teach” what one skilled in the art could not enable it to do – as expressly required under the case Microsoft cites (MS Br. at 46), Beckman Instruments, Inc. v. LKB Produktur AB, 892 F.2d 1547, 1551 (Fed. Cir. 1989) (Even a *patent*, to be invalidating prior art, “must enable one skilled in the art to make and use the apparatus or method”). No testimony was offered on how one skilled in the art would discover inherently in the thousands of printed pages that made up DX37 the need to change the fundamental structure of the VOBJF architecture, create a server code based on a protocol that never existed, and alter the Viola code to allow Viola to load an object. Wei never did it. A101173-78; A101203-05. Kelly never identified any of these “bugs” or the changes necessary in his expert report. A101307. Even the

two Sun engineers to whom Wei sent DX37 were unable to make Viola work at all, even though they presumably were skilled in the art. A101207; A170030.

Microsoft also contends that DX37 satisfies the claim limitations because, if it were to retrieve a page over some distributed hypermedia network environment, it supposedly would load the object associated with that page if that object were on the same local machine as the browser. MS Br. at 45 (citing A101284). But this contention ignores the threshold failure of DX37 to function with any known server, *i.e.*, with any existing distributed hypermedia network environment. Worse, Kelly's testimony, on which Microsoft relies, is unsupported and hypothetical. It entirely ignores the fact that this hypothetical situation plainly would require *new changes* to the VOBJF architecture. If the page were to be moved to a server, but the object were to remain local, the <VOBJF> tag in the page would have to be changed to include the file path for a separate machine from the one on which the page resided – the local machine where the object could be found. *No evidence was presented or demonstration conducted* that would support the bald assertion that such change would allow an actual, reduced-to-practice public version of Viola to achieve the requisite functionality.

In the end, Microsoft is left to rely on the uncorroborated testimony of Wei and the unsupported testimony of Kelly. Microsoft claims that Wei eventually “fixed”

DX37 to produce some Viola version (an “alpha” release) capable of use in a distributed hypermedia network environment. MS Br. at 47. Wei’s testimony cited by Microsoft says no such thing. A101162-65. In any event, no code from any such release was ever produced or offered in this case to corroborate this unsupported claim, and it was properly rejected. A101187-88; A101191-92; A101194-95; Finnigan Corp., 180 F.3d at 1367-69. As this Court explained, “post-invention oral testimony is more suspect, as there is more of a risk that the witness may have a litigation-inspired motive to corroborate” the invention. Sandt Tech., Ltd. v. Resco Metal & Plastics Corp., 264 F.3d 1344, 1351 (Fed. Cir. 2001).

Nor does Kelly’s conclusory testimony meet Microsoft’s burden:

Well, we’ve seen that this was a browser. It describes the browser for use on the World Wide Web with -- which is a network environment. And it’s a distributed hypermedia environment. It gets web pages. So it has all of these features.

A101277. But whether Viola could work as a “browser” on the “web” is immaterial; the issue is whether the VOBJF architecture works in a distributed hypermedia network environment with the particular functionality claimed by the ’906 patent. The Kelly conclusion does not demonstrate *that*.

This conclusory testimony is precisely what this Court has warned against:

For a prior art reference to anticipate a claim, the reference must disclose each and every element of the claim with sufficient clarity to prove its existence in the prior art. . . . Although this disclosure

requirement presupposes the knowledge of one skilled in the art of the claimed invention, that presumed knowledge does not grant a license to read into the prior art reference teachings that are not there. An expert's conclusory testimony, unsupported by the documentary evidence, cannot supplant the requirement of anticipatory disclosure in the prior art reference itself.

Motorola, Inc. v. Interdigital Tech. Corp., 121 F.3d 1461, 1473 (Fed. Cir. 1997). This Court has consistently rejected expert testimony that is in "the form of pure unsupported assertion," and it was proper for the court to do so here as well. W.L. Gore & Assocs., Inc. v. Garlock, Inc., 721 F.2d 1540, 1554 (Fed. Cir. 1983). See also Jamesbury Corp. v. Litton Indus. Prods., Inc., 756 F.2d 1556, 1563 (Fed. Cir. 1985) (granting JNOV where the record did not support an expert's conclusory testimony of anticipation).

b. *"type information . . . utilized by said browser to identify and locate an executable application"*

This claim element requires that the tasks of utilizing the type information to identify and locate the executable application "are performed by the browser." A000089. For this element, "as a factual matter, one must be able to characterize the browser as doing the heavy lifting." A000088.

There is simply no evidence from which a trier of fact could find this claim element in Viola. DX37 undisputedly relied on the "object," not the browser, to use "type information" to "identify and locate an executable application." In the VOBJF

architecture, the file path for the executable application (vplot) was hard coded into the Viola object (plot.v). A100844-47; A101202-03; A101309. This file path, according to Kelly, was “type information.” A101309. Moreover, Kelly agreed that this information gives “everything you need for the location of the application.” Id. Kelly repeatedly testified that once the file path has been determined, the executable application has been located. A101261-63; A101289-90; A101295.

Plot.v, which has the executable application’s file path, is not part of the browser. A101309. Kelly unequivocally agreed that “plot.v is the object, *not the browser.*” Id. Thus, it is the plot.v object that, by specifying the file path, fully identifies and locates the executable application.

Faced with these admissions, Microsoft argues that Kelly testified that Viola “taught” this element. A101278. Kelly’s conclusory testimony is at odds with the evidence. His own admission is that it is the object that contains the hard-coded file path to the executable application, and that is all that is necessary to locate the executable application.

c. *“browser application to parse a first distributed hypermedia document”*

Another ’906 claim element requires that the browser “parse a first distributed hypermedia document” – something which Viola alone could not do. Both Wei and Kelly testified that SGMLS (software that Wei did not write but downloaded from

somewhere else) parsed the hypermedia document. A101175-76; A101180-82; A101184; A101204-05; A101283; A101293. Both admitted that the SGMLS program was not part of Viola, or included on the distribution that makes up the Viola code on DX37. Id. Wei admitted that there was no place a member of the public could go in this time frame to know that they needed to download an SGMLS parser to add to Viola. A101182. There is no evidence that any other parser would work with the Viola system, or how it would work, or whether a 1993 parser would have worked in such a system.

2. DX34 does not anticipate

Microsoft had a full opportunity to declare the full use it planned to make of DX34. Cross-examination of Wei on voir dire showed undisputably that DX34, like DX37, lacks elements of the '906 invention. Quite apart from the other § 102(b) and § 102(g) grounds on which the district court excluded DX34, the failure of DX34 to meet all claim elements independently supports JMOL against Microsoft's sole contention based on DX34 – that it is the “invention” invalidating the '906 patent under § 102(b) and § 102(g).

DX34 is not a computer program product for use in a distributed hypermedia network environment. It was never used in one. A101176; A101178. Like DX37, it was not capable of such use because the VOBJF architecture was geared for use on

one machine, A101164; A101176-77, and it did not operate with any server protocol in existence. A101177-78; A101186. In DX34, like DX37, the “object,” not the browser, used “type information” to “identify and locate” an executable application – as acknowledged by Microsoft’s counsel when he characterized plot.v as “contain[ing] the location of an application or the name of the application.” A101164. Finally, DX34 does not “parse,” but instead uses the separate SGMLS program to do so. A101175-76; A101180-82.⁶

Accordingly, this Court need not reach the remaining issues related to invalidity because Microsoft offered no clear and convincing evidence that either DX34 or DX37 met each element of the ’906 claims.

B. DX34 and DX37 are not § 102(b) prior art because they were not in “public use”

Even if DX34 or DX37 met all the claim limitations, they were not invalidating prior art under § 102(b) because Wei failed the essential requirement for public use: He did not put the “invention” in the public domain.

⁶ Not only is this an alternative ground of affirmance, it demonstrates that Microsoft cannot establish prejudice based on the exclusion of DX34. Fed. R. Civ. P. 61; see ATD Corp. v. Lydall, Inc., 159 F.3d 534, 549-50 (Fed. Cir. 1998); see also Deere & Co. v. Int’l Harvester Co., 710 F.2d 1551, 1557-58 (Fed. Cir. 1983). Indeed, Microsoft itself has repeatedly represented that DX34 and DX37 were “not different in any material respect.” A101127; A101171; MS Br. at 37 n.15; 37-38; 45. Thus, even if DX34 would have been admitted, it did not provide evidence sufficient to send Microsoft’s invalidity case to the jury.

1. As to DX34, on which Microsoft principally relies, the facts underlying the district court's conclusion of no "public use" are not in dispute. Microsoft relies on the single demonstration on May 7, 1993, the only demonstration of DX34 at any time. A101176. Neither the Sun engineers nor any other member of the public ever saw or were given access to the DX34 code of the demonstration. A101174-75. The demonstration took place entirely on one machine. A101176-77.

Because DX34 was not demonstrated in a distributed hypermedia network environment, and the code was never given to anyone, the demonstration did not make public the one essential piece of functionality unique to DX34 that Microsoft claims would allow DX34 to be used in a distributed hypermedia network environment: The alleged ability to load an object retrieved over the Internet (which it could not be, given the Viola structure and server protocol issues). A101161-64; A101173-75; A101203-05. And there was no corroborated evidence, much less clear and convincing evidence, that this functionality – even if present in DX34 – was ever demonstrated publicly or, indeed, ever appeared again in Viola after DX34. *Id.*; A101176-77.

The Sun demonstration did not serve to put into the "public" domain this feature of DX34. W. L. Gore, 721 F.2d at 1549-50. In W. L. Gore, this Court held that use by a party other than an inventor of a machine did not constitute a public

disclosure of a patented process because looking at the machine would not reveal anything regarding the process the machine is being used to practice. Id.; cf. Laporte v. Norfolk Dredging Co., 787 F.2d 1577, 1582 (Fed. Cir. 1986) (distinguishing W. L. Gore because “invention was discoverable from device”). The same principle applies here: Even the Sun engineers could not have learned the critical feature from the demonstration.

This Court has long distinguished between the bar of § 102(b) when applied to an inventor, and when applied to the asserted prior use of one other than the inventor. “[W]hen an asserted prior use is not that of the applicant, § 102(b) is not a bar when that prior use or knowledge is not available to the public.” Woodland Trust, 148 F.3d at 1371. In such a circumstance, the law requires that the use disclose to the public a prior invention in a way that will allow a later inventor to obtain knowledge of that invention. W.L. Gore, 721 F.2d at 1550.

Microsoft’s reliance on Netscape Communications Corp. v. Konrad, 295 F.3d 1315 (Fed. Cir. 2002), is therefore misplaced. In Netscape Communications Corp., and the other cases cited by Microsoft, the public use bar was applied to the prior use of an *inventor*, and thus it “serv[ed] the policies of the patent system, for it encourages prompt filing of patent applications.” Id. at 1320 (quoting Allied Colloids v. Am. Cyanamid Co., 64 F.3d 1570, 1574 (Fed. Cir. 1995)).

2. The DX37 browser also was not in public use. When Wei sent DX37 to the Sun engineers in late May 1993, to whom he had earlier in May given a partial demonstration of DX34, he deliberately and successfully undertook steps to ensure that only the Sun engineers received the code, and that members of the public were precluded from access to it. A101207; A101216-17; A151665; A170024. Moreover, there is no evidence that Sun was able to make DX37 function at all, or as required by the '906 claims. A101207; A170030.

3. Wei asserted that, after May 1993, he sent a so-called "alpha" release to three individuals. A101206-08; A101211; A151531-32; A170035-36. But Microsoft utterly failed to prove what was in this code. The recipients did not testify. None of this code was produced or offered. The content of the code and the VOBJF functionality is unknown. Whatever was given, it was *not* Viola as it existed on either DX34 or DX37 – Wei made changes to the Viola code during the period of these distributions, but did not maintain records of the specific versions distributed. A101208. In any event, these distributions were also made under circumstances that did not place the Viola code, if it even was DX37, in "public use." These distributions were made in such a way that it was "nearly impossible" for the public to get Viola. A101211.

Likewise, there is no corroboration of the functionality of the code during Wei's demonstration at the World Wide Web Wizard's Workshop in 1993. A101165; A101200; A101214; A101216. In fact, Wei did not even remember demonstrating an "embedded application." A101165. Moreover, it is undisputed that this demonstration was on one machine, not over a distributed hypermedia network environment. A101165; A101200; A101214.

The district court properly rejected Wei's uncorroborated testimony of the functionality of any Viola version distributed or used beyond DX34 and DX37. A101187-88; A101191-92; A101194-95. The general rule is that "oral testimony of prior public use must be corroborated in order to invalidate a patent." Juicy Whip, Inc. v. Orange Bang, Inc., 292 F.3d 728, 737-38 (Fed. Cir. 2002). "[C]orroboration is required of any witness whose testimony alone is asserted to invalidate a patent, regardless of his or her level of interest." Finnigan Corp., 180 F.3d at 1369.

Thus, Microsoft failed to prove that any Viola browser with the allegedly invalidating VOBJF functionality was ever in the public domain to invalidate the '906 claims under § 102(b).

C. Both DX34 and the VOBJF architecture were abandoned, suppressed, and concealed under § 102(g)

This Court recognizes two types of abandonment, suppression, or concealment. The first is implicated when an inventor actively abandons, suppresses, or conceals

his invention from the public. Fujikawa v. Wattanasin, 93 F.3d 1559, 1567 (Fed. Cir. 1996). The second occurs when abandonment, suppression, or concealment may be inferred from the prior inventor's unreasonable delay in making the invention publicly known. Dow Chem. Co. v. Astro-Valcour, Inc., 267 F.3d 1334, 1342 (Fed. Cir. 2001). Assuming *arguendo* that Microsoft had proven the DX34 and DX37 browsers met all the '906 claim limitations, Plaintiffs-Appellees then had the burden of *production* to offer evidence sufficient to create a genuine issue of material fact as to whether the prior inventor abandoned, suppressed, or concealed the invention. Apotex, 254 F.3d at 1037-38. Microsoft then had the burden of *persuasion* to rebut that evidence with clear and convincing evidence to the contrary. Id.

Plaintiffs-Appellees offered the required evidence – for both forms of abandonment, suppression, and concealment – and Microsoft failed to rebut it.

1. DX34 was abandoned suppressed, and concealed. Wei changed the functionality of DX34 through which Microsoft contends that version met the claim limitation of being capable of use in a distributed hypermedia network environment (notwithstanding the Viola structure, server issue, and the other missing elements). This supposedly key code was changed in creating the DX37 browser, and there is no evidence that the functionality of the VOBJF architecture as it existed in DX34 ever appeared again, outside of Wei's uncorroborated testimony. A101161-64;

A101173-75; A101203-05. Even when it did exist in DX34, Wei did not show or reveal it to the Sun engineers in the one demonstration Microsoft argues was public. A101174-77.

2. Wei abandoned, suppressed, and concealed the VOBJF architecture in its entirety – on which Microsoft’s invalidity arguments for both DX34 and DX37 were based – by January 17, 1994. A101172-73; A101175; A101184. By the time of the so-called “beta” release in February 1994, Wei had switched instead to an architecture based on a <LINK> tag which had different capabilities and for which no evidence that it met all of the ’906 claim elements outside Wei’s uncorroborated testimony was introduced. Id.

Microsoft asserts that Wei merely changed the name of the architecture and that support for the VOBJF architecture remained in other Viola versions. But, Microsoft simply offered no support for that assertion meeting the established requirements of corroboration and clear and convincing evidence. Sandt Tech., Ltd., 264 F.3d at 1350. The allegedly corroborating “evidence” cited by Microsoft either is not contemporaneous with the claimed functionality of the VOBJF architecture, is not tied to any particular code, or does not discuss the functionality of the VOBJF architecture. A101172-73; A101186-93; A170003-07; A170037-43; A170046-77. The district court properly excluded this evidence under F.R.E. 402 and 403.

A100457-59; A100466-73; A101186-93; A102019-26. This was not an abuse of discretion. See Kolmes, 107 F.3d at 1542.

In the end, this case vividly illustrates the importance of corroboration when someone claims under § 102(g) to have antedated the patentee's invention. As the district court noted on more than one occasion, Wei provided varying testimony on exactly what he claimed to have invented, when he claimed to have invented it, and whether it was publicly available. A101187-88; A101191-92; A101194-95. The only evidence constituting corroboration (the code itself), however, puts beyond dispute that neither DX34, DX37, nor any other Viola browser supports an invalidity defense under § 102(g).

D. Microsoft did not present an obviousness case

As the district court noted, “[Microsoft’s] arguments are almost as much of a moving target as Viola.” A101188. This is particularly true regarding its obviousness defense.

Before trial, Kelly offered an opinion exclusively on anticipation, admitting that he did not perform a Graham analysis. A101190. Nonetheless, before putting Wei or its expert on before the jury, Microsoft unequivocally stated that it was pursuing a case of obviousness with DX37: “The argument we want to make to the jury about Defendant’s Exhibit 37 is *not that it anticipates*. . . . With respect to this

exhibit it's *obviousness*." A101189 (also admitting that with DX37, Microsoft is gap-filling under Scripps).

Despite these representations, Microsoft presented testimony purporting to show that DX37 contains *each element* of the '906 invention, and thus anticipates. E.g., A101255; A101282. Kelly testified that he believed that the '906 patent was invalid exclusively because of *anticipation*:

Q. . . . First of all, so that the ladies and gentleman of the jury understand what it is we're talking about, it's your opinion that claims one and six of the '906 patent are invalid for anticipation, correct?

A. That's quite correct, yes.

Q. And anticipation means that each and every element of the claim is found in the prior art, right?

A. That's correct.

A101306.

As shown above, Kelly and Wei made concessions confirming that elements were missing from this "anticipatory" reference. Only then – long after Microsoft's witnesses had left the stand and evidence was closed – did Microsoft return to its original obviousness contention to try to defeat Plaintiffs-Appellees' JMOL motion. When pushed by the district court, Microsoft's counsel admitted that its *post hoc* obviousness case was no different from its anticipation case, and argued – without any specifics – that although Kelly testified that there were no gaps, the gaps exposed

by Plaintiffs-Appellees could be supplied elsewhere. A101354-57. Microsoft's counsel argued: "[I]f the jury finds that they are missing elements, then it would have been obvious to supply them based on the teachings of that reference given the knowledge of a person of ordinary skill." A101355.

An after-the-fact statement by counsel, however, does not create an obviousness defense. The obviousness determination is a disciplined endeavor. The guideposts in Graham and Panduit protect against determinations of obviousness made on the impermissible basis of hindsight. ATD Corp., 159 F.3d at 546. Microsoft had the burden to offer a prima facie case of obviousness, and to establish by clear and convincing evidence the factual foundation of its invalidity claim. Ashland Oil, Inc. v. Delta Resins & Refractories, Inc., 776 F.2d 281, 291, 293 (Fed. Cir. 1985).

Microsoft wholly failed to provide evidence on (1) the scope and content of the prior art; (2) the differences between the prior art and the invention; or (3) explain and prove the existence of a suggestion or motivation to combine. See ATD Corp., 159 F.3d at 546. In the absence of such evidence, the district court correctly granted JMOL.

III. THE RULING ON INEQUITABLE CONDUCT WAS CORRECT

The district court correctly held that there was no inequitable conduct in Doyle's alleged failure to disclose. Microsoft did not meet its burden to provide "clear and convincing proof of: (1) prior art or information that is material; (2) knowledge chargeable to applicant of that prior art or information and of its materiality; and (3) failure of the applicant to disclose the art or information resulting from an intent to mislead the PTO." FMC Corp. v. Manitowoc Co., Inc., 835 F.2d 1411, 1415 (Fed. Cir. 1987). A mere showing that "art or information having some degree of materiality was not disclosed" is insufficient. Id.

A trial court's decision on inequitable conduct will be reversed only if it rests on clearly erroneous findings of fact, a misapplication or misinterpretation of applicable law, or if it evidences a clear error in judgment. Norian Corp. v. Stryker Corp., 363 F.3d 1321, 1331 (Fed. Cir. 2004). "Clear error" requires a "definite and firm conviction that a mistake has been committed." Molins PLC v. Textron, Inc., 48 F.3d 1172, 1178 (Fed. Cir. 1995). That standard is not met here.

First, the alleged prior art was not material. Microsoft's entire argument is based on its view that the district court had a "fundamental misconception that infected its other Viola rulings." MS Br. at 49. Since, as demonstrated above, the

court's rulings on Viola were legally and factually correct, Microsoft's arguments necessarily fail.

What information Doyle had was not material. Microsoft does not dispute that Doyle did not have access to or have either DX34 or DX37, any other pre-1994 Viola code, or any other documents or information that corroborate Wei's claims of the functionality he supposedly demonstrated in 1993. A000033-34. Microsoft does not challenge the finding that the '906 invention date was no later than January 27, 1994. A000035. Nor does Microsoft challenge the district court's finding that all of the information Doyle had in his possession involved versions of Viola that post-dated the '906 invention date. A000033.

Contrary to Microsoft's assertions, the district court did not view materiality only by examining DX34 and DX37 in isolation, divorced from other evidence. In fact, the court examined individually the disparate pieces of Viola information and clearly discussed the "entirety" of them. See A000028-38. What the court correctly recognized is that only DX34 and DX37 (the only Viola code offered), demonstrated what functionality Viola achieved (code to which Doyle did not have access to or know existed), and no other printed publication offered by Microsoft revealed the functionality of Viola or the implementation of the <VOBJF> tag. A000033-34.

No expert or other witness testified that any exhibit regarding Viola in Doyle's possession was material to the patentability of any claim under consideration in the '906 prosecution, either alone or in combination with any other reference. Therefore, the district court's determination that Doyle could have disclosed "nothing more than Wei's assertion that he achieved something" was proper. A000037. Microsoft cites no law requiring such disclosure.

Second, the district court correctly concluded that the applicants did not believe that Viola was material, and they had no intent to mislead the PTO. These findings were supported by the record. Based on the factual findings discussed above, the court concluded that Doyle did not:

[M]ake a tactical decision to forego disclosure because of the advantages of defending his invention after patent issued. He made the decision because he believed that Wei was full of hot air and he had nothing concrete to display to an Examiner. So I credit his testimony that he had no intent to deceive the Examiner.

A000041.

This is not the case that Microsoft posits – a case where the patentee knew or should have known of the materiality of "prior art" – let alone one where an inference of intentional deception is compelled. No case cited by Microsoft involves the situation here, where a patentee did not know, and could not reasonably learn, what a *claimant* was actually up to. The district court's factual and credibility findings on

the spottiness and unreliability of any knowledge of what Wei was *actually up to* is ample basis for the court's conclusion.

Significantly, where, as here, a trial court has made extensive factual findings which are supported by the record, and a determination based on demeanor and testimony that a witness is credible, high deference must be given to the trial court. This Court has stated that "credibility determinations by a trial judge can virtually never be clear error." Monon Corp. v. Stoughton Trailers, Inc., 239 F.3d 1253, 1263 (Fed. Cir. 2001) (citation omitted).

When distilled, Microsoft's entire argument rests upon challenging the district court's factual and credibility determinations. Microsoft would have this Court reverse credibility determinations because the district court took issue with Doyle's answer on one or two collateral points. MS Br. at 49. The district court itself, however, explained why that disagreement did not undermine the express findings as to Doyle's good faith on the dispositive matter. "Doyle may have convinced himself of the truth of everything he said. And, even basically honest persons rewrite history on matters they believe are not central to a dispute." A000040. On the critical issues of knowledge of materiality and intent to deceive, the district court specifically found Doyle to be truthful. A000039-41.

Those determinations should not be disturbed on appeal.

IV. THE DISTRICT COURT COMMITTED NO ERROR OF CLAIM CONSTRUCTION

The district court properly construed the claim language in light of its ordinary meaning and the intrinsic evidence and gave appropriate jury instructions.

A. Executable applications are not limited to standalone applications

The district court determined that “‘executable application,’ as used in the ’906 Patent, is any computer program code, that is not the operating system or a utility, that is launched to enable an end-user to directly interact with the data.” A000089. Microsoft asks this Court to import a limitation into this construction, arguing that an “executable application” is limited to “standalone” programs, allegedly thereby excluding DLL-based applications like Microsoft’s Active X controls. The district court correctly rejected this argument.

1. This Court has repeatedly stated that “[g]eneral descriptive terms will ordinarily be given their full meaning,” and that “modifiers will not be added to broad terms standing alone.” Johnson Worldwide Assocs., Inc. v. Zebco Corp., 175 F.3d 985, 989 (Fed. Cir. 1999). It is improper to import unstated and unintended words or limitations into the claims where no such words or limitations exist. Karlin Tech. Inc. v. Surgical Dynamics, Inc., 177 F.3d 968, 971 (Fed. Cir. 1999).

Contrary to Microsoft, the “ordinary meaning” of the term “application,” by itself, does not exclude DLLs. To argue otherwise, Microsoft must rely on selective

excerpts from one of many versions of its own computer dictionary. MS Br. at 51-52.

The definition of “application” that it cites is very broad: “A computer program designed to help people perform a certain type of work,” different from:

[A]n operating system (which runs a computer), a utility (which performs maintenance or general-purpose chores), and a language (with which computer programs are created). Depending on the work for which it was designed, an application can manipulate text, numbers, graphics, or a combination of these elements.

Microsoft Press Computer Dictionary 23-24 (2d ed. 1994). Nothing about this definition excludes DLLs from constituting the application. Indeed, Microsoft’s own expert on claim construction admitted this very point:

Q. . . . [O]ne of ordinary skill in the art as of the date of filing of the ’906 patent would necessarily understand that a spreadsheet, a word processor, a database program could be built out of DLLs, isn’t that correct sir?

A. That is correct.

A100255; see also A100251 (agreeing that one can “build an application with DLLs”); A100252-54. At trial, Kelly admitted that Microsoft’s Active X controls, which are DLLs, are “executable applications” as that term is used in the ’906 patent. A101306.

Microsoft attempts to avoid this conclusion by pointing to a definition of “computer program” that says that the term “usually implies a self-contained entity, as opposed to a routine or a library.” Microsoft Press Computer Dictionary 90 (2d

ed. 1994). But “usually” on its face is not enough to exclude, particularly where what is “usually” the case *in general* is not “usually” the case *in the present context* (as made clear from the specification). See TI Group Auto. Sys., Inc. v. VDO N. Am., L.L.C., No. 02-1630, 2004 U.S. App. LEXIS 13445, at *15-17 (Fed. Cir. June 30, 2004) (all applicable dictionary definitions embraced); Nystrom v. Trex Co., Inc., No. 03-1092, 2004 U.S. App. LEXIS 13407, at *14 n.2 (Fed. Cir. June 28, 2004) (“claim terms may be construed to encompass all dictionary definitions not inconsistent with the intrinsic record”); Brookhill-Wilk 1, LLC v. Intuitive Surgical, Inc., 334 F.3d 1294, 1300 (Fed. Cir. 2003). Other versions of Microsoft’s dictionary offer broader understandings. Microsoft Press Computer Dictionary 27, 384 (3d ed. 1997) (defining “application” by reference to “program,” which is defined as “[a] sequence of instructions that can be executed by a computer. . . . software”). The district court cited still other broad dictionary definitions. A000065.

Microsoft’s effort to rely on a dictionary’s definition of “executable” also fails. It makes the specious assertion that a definition of “executable program” as “ready to run” excludes non-standalone applications, which Microsoft says are “not ready to run.” MS Br. at 52. But the very definition of “executable program” Microsoft cites goes on to describe the meaning of “ready to run” as simply this: “[T]he user does not have to alter the program in any way before being able to run it.” Microsoft

Press Computer Dictionary 153 (2d ed. 1994). That understanding in no way carves out non-standalone applications. Nor do other contemporaneous dictionary definitions of “executable program.” See, e.g., IBM Dictionary of Computing 250 (10th ed. 1994) (“[t]he set of machine language instructions that constitute the output from the compilation of a source program. . . .”).

Here, it is particularly unnatural to restrict the claim phrase to “standalone” applications, because the gist of a standalone program is – for lack of better words – to function on its own. In the context of the ’906 claims, however, the “executable application” is functioning only under the control of the browser (automatically invoking application in order to enable interactive processing in “first browser-controlled window”). Since the executable application is being controlled by the browser, whether that application can function in another environment as a standalone is not germane to the ’906 patent.

2. As the district court noted, the specification does not limit executable application to standalone programs. A000069-70. The specification states: “[A]ny manner of applications program may be specified by the TYPE element so that other types of applications, such as a spreadsheet program, database program, word processor, etc. may be used with the present invention.” A150018. Whether these examples are “standalone” programs or modules is irrelevant to implementation of

the invention. As long as they can be automatically invoked to allow interactive processing within the displayed Web page, they fall within the claim term.

Furthermore, the specification notes that *all* methods of implementation of the “browser application” are possible – including *non-standalone programs* such as “routines, processes, subroutines, modules, etc.” A150018. The very same breadth and flexibility equally should apply to the “executable application.”

Nor, contrary to Microsoft’s contention, does the preferred embodiment limit executable applications to standalone applications. It is axiomatic that claims are not limited to preferred embodiments. Liebel-Flarsheim Co. v. Medrad, Inc., 358 F.3d 898, 906 (Fed. Cir. 2004); Modine Mfg. Co. v. U.S. Int’l Trade Comm’n, 75 F.3d 1545, 1551 (Fed. Cir.), cert. denied, 518 U.S. 1005 (1996). In the ’906 patent the inventors unequivocally stated “[m]any such changes or modifications will be readily apparent to one of ordinary skill in the art. *The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.*” A150019 (emphasis added); see also A150015-17 (“In one application”; “Other applications of the invention are possible”; “Another embodiment of the present invention uses”; “Other applications of the invention are possible”; “Another type of possible application of this invention”).

The specification is clear – many embodiments are possible, and the role of the executable application makes any “standalone” restrictions wholly artificial. Karlin Tech., Inc., 177 F.3d at 973. Against that background, there certainly is no “clear disclaimer” in the specification that would limit the invention to standalone executable applications. Liebel-Flarsheim Co., 358 F.3d at 908; Brookhill-Wilk 1, LLC, 334 F.3d at 1298.

3. The prosecution history clearly confirms that the applicants contemplated that “executable application” would encompass *any* executable code that allows interactive processing of the object. Liebel-Flarsheim Co., 358 F.3d at 913; see also Honeywell Inc. v. Victor Co. of Japan, Ltd., 298 F.3d 1317, 1323-24 (Fed. Cir. 2002). In the very first office action, the examiner defined the executable application in this way: “The disclosed prior art does not have embedded controllable application [*executable/interpretable/‘launchable’ program instructions/codes*] in the hypermedia document].” A150083 (emphasis added). As the district court recognized, “[t]his is a broad definition that does not exclude components” A000072.

Furthermore, the applicants, in overcoming prior art, consistently noted that Microsoft’s Active X controls specifically incorporated the features of the invention. A150349; A150371-72. They also cited a specific reference, Denning, that describes

how Microsoft practices the invention through Active X and DLLs. A150918-22. No careful reader of the prosecution history could mistake the *inclusion* of Active X's DLLs.

4. Contrary to Microsoft's argument, nothing in the prosecution history effects a clear and unmistakable disavowal of DLLs as a class. Anchor Wall Sys., Inc. v. Rockwood Retaining Walls, Inc., 340 F.3d 1298, 1310 (Fed. Cir. 2003). Microsoft's argument is premised upon the mistaken belief that by simply outlining how a particular piece of prior art operates, an applicant somehow has expressly disclaimed coverage of *all aspects* of the prior art, even if the actual grounds for distinction are narrower. The district court rejected this argument as "on too high a level of abstraction" (A00073) and concluded that the applicants "did not say their invention used DLLs. Instead, the applicants said that [*the prior art*] DLLs did not do the same thing as their invention." A000076 (emphasis added). Distinguishing *particular* prior DLLs in *particular* respects did not disclaim DLLs as a class – Microsoft's only argument for disclaimer of the Active X DLLs. N. Telecom Ltd. v. Samsung Elecs. Co., Ltd., 215 F.3d 1281, 1293-95 (Fed. Cir. 2000); Karlin Tech. Inc., 177 F.3d at 974.

When the applicants distinguished prior art, they always distinguished *specific usages* of DLLs in the prior art, not DLLs as a class of executable applications. The

examiner understood the distinction. The reason, as he stated, is that the “crux” of the ’906 invention is parsing of an embed text format by a browser to “automatic[ally] invoke an external application to provide interactive control.” A150520.

Thus, the applicants did not distinguish all use of DLLs when distinguishing the Khoyi reference (which describes Microsoft’s OLE, *not* Active X, technology). The applicants distinguished Khoyi on the grounds that the display of the data object provided by the Khoyi functionality was not interactive, and in the ’906 invention such interactivity could be achieved “without requiring Khoyi-like capabilities in the operating system.” A150340; A150342-43. Furthermore, in a PTO affidavit, Doyle specifically noted the distinction between Khoyi-OLE technology and Active X, stating that Active X incorporated the features of the invention, including automatic invocation of an external executable application to allow interactive processing. A150349; A150371-72 (quoting a Microsoft manager that Active X is not just simply OLE for the Internet).

Later in the prosecution, the examiner rejected the claims based on a combination of the Mosaic Web browser and Koppolu (which describes Microsoft’s then-existing OLE facility), arguing that it would have been obvious to combine the Mosaic browser with the object handler shown by Koppolu (DLL code that can be automatically invoked at document-rendering time and that allowed, for example, a

spreadsheet to be included in a Word-based document) to arrive at the '906 invention. A150390-92.

But, again, the applicants distinguished Koppolu on grounds other than DLLs as a class. The applicants specifically pointed out that the *particular kind* of object handlers of Koppolu did not allow *interactive processing* of the object, since in the Koppolu embodiment such editing of the object was not possible. A150905-07.

In allowing the claims, the examiner agreed with the applicants' arguments, noting that:

The examiner agrees that the claimed external executable application is not a code library extension nor object handler (e.g. windows dll and OLE) *as pointed out in applicant's argument.* (Paper #19 pages 12-14).

A151030-31 (emphasis added) (citing A150905-07). Thus, even if the *examiner's* statement could constitute a disclaimer by the *applicants* (which it cannot, ACCO Brands, Inc. v. Micro Sec. Devices, Inc., 346 F.3d 1075, 1079 (Fed. Cir. 2003)), the examiner did not suggest (as Microsoft argues) that DLLs could never be an executable application. Rather, he addressed (and accepted) only the applicants' argument that the *particular kind* of object handlers as taught in Koppolu did not render the invention obvious.

B. The browser's use of outside resources is not in dispute

The district court concluded that the task of utilizing type information to identify and locate the executable application must be “performed by the browser. This is a fact-intensive inquiry.” A000089. The court noted that in the preferred embodiment, “the browser does not work alone. The specification makes clear that the inventors contemplated the browser’s use of some outside resources.” A000086. In reaching its construction, the court noted that even “Microsoft does not propose a claim construction that would entirely preclude the browser from using the operating system or some external resource.” A000086; see also A000088. The court’s construction was correct, and the jury instruction given on this issue was proper.

1. The construction was correct. The claim language requires a browser application that identifies and locates an executable application by utilizing the type information associated with the object. A150020. The specification describes a preferred embodiment in which the browser determines which application to run after consultation with an operating system resource, a database outside the browser itself. A150019 (discussing use of outside resources to help determine applications).

Microsoft’s construction – which would exclude involvement of the operating system, and require that the browser identify a file name and file path – would read

the preferred embodiment out of the '906 claims. Such constructions are “rarely, if ever, correct and . . . require highly persuasive evidentiary support.” Vitronics Corp. v. Conceptronic, Inc., 90 F.3d 1576, 1583 (Fed. Cir. 1996). See also Amgen Inc. v. Hoechst Marion Roussel, Inc., 314 F.3d 1313, 1349 (Fed. Cir. 2003). Here, there is no support – much less “highly persuasive” support – to overcome the usual presumption that the claims cover the preferred embodiment.

The applicants did not disclaim any operating system role in the prosecution history. As with the term “executable application,” Microsoft’s argument improperly overlooks the specific bases on which the applicants distinguished their invention from the prior art. N. Telecom, 215 F.3d at 1294.

Microsoft contends that, because the inventors stated that their invention was not obvious in light of the proposed combination of the Mosaic browser and the Khoyi operating system, they gave up a construction of the term “utilized by said browser” that would allow the browser to use the operating system. This is a non-sequitur. If, by arguing against the proposed combination of Mosaic and Khoyi, the inventors gave up implementation of the invention by any use of the operating system (Khoyi), the inventors must equally have given up any implementation of the invention that used a browser (Mosaic).

In fact, the applicants argued that, unlike Khoyi, their invention did not require special capabilities of the operating system to provide *interactivity*, and therefore could be implemented on a wider variety of operating systems. A150340; see also A150342-43. The distinctions the applicants drew over Khoyi had nothing to do with whether or not the browser could consult operating system resources to determine which application to launch.

Microsoft selectively quotes a passage in which the inventors point out – merely as background information – that Koppolu uses the underlying operating system registry to locate the appropriate application. MS Br. at 14-15; 57 (citing A150902). Based on this, Microsoft argues that the applicants disclaimed any use of the operating system by the browser.

Microsoft’s argument is legally incorrect. Claims are not limited by patentees’ full description of prior art where the statements do not relate to the actual point of distinction during the patent prosecution (lest full descriptions for the examiner’s benefit be discouraged). N. Telecom, 215 F.3d at 1293-94. It is also factually incorrect. The actual focus of the quoted passage is, as the inventors say in the topic sentence of the quoted paragraph, that “OLE does not **parse** text tags in the document in order to render the document (as required by the Applicants’ claims).” A150902 (bold in original). In omitting this sentence from its quotation (MS Br. at 14-15; 57),

Microsoft creates the illusion that the applicants were making an argument that they were not. The applicants distinguished Koppolu on the ground that it could not *parse text formats* (a critical part of the '906 patent), not that the browser needed to find the executable application without any assistance from other files or the operating system.

2. Microsoft cannot establish prejudicial error in the instruction given to the jury about this claim language. See Ecolab Inc. v. Paraclipse, Inc., 285 F.3d 1362, 1373 (Fed. Cir. 2002); Teleflex, Inc., 299 F.3d at 1328.

Microsoft takes issue with one sentence of the jury instruction that said “[t]he inventors contemplated the browser’s use of some outside resources such as the operating system.” A101463. Microsoft’s sole complaint is that this sentence left the jury without guidance on the proper role of the browser vis-a-vis the operating system. MS Br. at 58-59.

But the next sentence of the jury instruction provided this guidance: “It must be the browser, not the operating system, that must do the heavy lifting of identifying and locating the executable application.” A101463. This was language suggested by *Microsoft* at the Markman hearing: “The heavy lifting, identifying and locating is what the browser has to do. That is what we are saying, it is the browser, not the operating system, that does that.” A100313; see also A000088. At the jury

instruction conference, Microsoft's counsel stated, in reference to "heavy lifting" in the jury instruction that: "We like it at this point, Your Honor." A101408.⁷

With the guidance adequate, Microsoft is left with an empty complaint about the sentence. It never challenged the sentence as incorrect: It told the district court that "it's undisputable, but its also immaterial." A101407. Nor could it have argued incorrectness, because throughout this case, Microsoft agreed that the '906 invention contemplates that the operating system is involved at some level. A000086; A001737. Indeed, when Microsoft proposed alternative clarifying language – "The browser may go to the computer storage, find an outside resource such as the MIME database and consult it" (A002308-09) – it used language not dissimilar to the language given to the jury.

In short, because the instruction as a whole properly guided the jury, the claim construction should be affirmed. See Delta-X Corp. v. Baker Hughes Prod. Tools, Inc., 984 F.2d 410, 415 (Fed. Cir. 1993).

3. Even if the claim were construed as Microsoft now contends, to require the browser actually to identify the file name and file path of the executable application, Microsoft could not demonstrate that the infringement result could be any

⁷ Felten testified in detail that the process undertaken by the browser is in fact the "heavy lifting" in linking the type information in the Web page to the executable application. A100834-39. Microsoft does not appeal infringement under the district court's construction.

different. Throughout this litigation, Microsoft's position was that it did not infringe because Internet Explorer did not determine the file name or file path of the executable application. MS Br. at 60; A101290; A101445-46. But that position was conclusively proven false at trial, making the claim construction dispute immaterial.

Mr. Wallent, Microsoft's General Manager for the Windows Client Platform team, admitted at trial that Internet Explorer, when identifying and locating the executable application, does obtain the file name and file path for that executable application:

Q. The code we're looking at here, sir, is actually capable of obtaining the file name and file path for an Active X control, isn't it?

A. Yes.

Q. And this is browser code that can do this, right?

A. Yes.

A101123-24; A101301.

Microsoft's only response to this compelling admission is that the browser finds the file name and file path of the application for another purpose. The evidence, however, is to the contrary. The purpose for obtaining the file name and file path is to determine whether the desired application is already located on the local machine – in other words, to identify and locate the executable application. A101301. If no executable application is found locally, or if one is found with an unsuitable version

number, then the browser downloads a new executable application from a location determined from the CODEBASE type information, which, in the website example was a URL. *Id.*; A100835-36; A101121.⁸

V. MICROSOFT HAS LIABILITY FOR ITS EXPORT SALES UNDER § 271(F)

The district court properly held that Microsoft faced liability for its export sales under 35 U.S.C. § 271(f). The facts are not in dispute. Microsoft conceives, writes, compiles, tests, debugs, and creates a master version of the accused products in Redmond, Washington. The Windows code is assembled and operational in its final form in the United States. Microsoft sends that code on a “golden master” in pre-install kits to OEMs. The OEM “uses the [pre-install] kit to install the Windows code on a particular computer-useable medium such as a disk, CD, or computer hard drive. . . . The foreign OEM then installs the Windows Code, now on a computer-useable medium made by the foreign OEM, into each computer product which is sold abroad.” A003094-95.

⁸ This Court has construed the term “URL” as “something that identifies the location of relevant information segments.” *ACTV, Inc. v. Walt Disney Co.*, 346 F.3d 1082, 1090 (Fed. Cir. 2003). This is consistent with the ’906 specification which describes the “location” of an object with a URL. A150013-14; A150018. Thus, there is no basis for Microsoft’s argument that “identify and locate” is limited to a file name and file path.

There is no dispute that the “golden master,” created here, is then exported abroad, as required by § 271(f). Pellegrini v. Analog Devices, Inc., No. 04-1054, 2004 U.S. App. LEXIS 14017, at *10 (Fed. Cir. July 8, 2004). In dispute here is whether Microsoft, when it exports its “golden master” to foreign OEMs, supplies from the United States “all or a substantial portion of the *components* of a patented invention . . . in such a manner as to actively induce the *combination* of such components . . .” 35 U.S.C. § 271(f)(1); or “any *component* of a patented invention . . . intending that such component will be *combined*. . . .” 35 U.S.C. § 271(f)(2) (emphasis added).

Here, the “patented invention” is a “computer program product.” A150020. The district court properly held that the Windows code on the golden master is a “component” of that invention, and that when the code is installed on computers by OEMs, there has been a “combination” that would infringe if it occurred in the United States. A000051-54.

At the core of Microsoft’s argument is its contention that software cannot be a “component” of a patented invention because it is intangible. Nothing in the text of § 271(f) imposes a requirement of “tangibility” on a component. Undefined terms in a statute are deemed to have their ordinarily understood meaning. Int’l Bus. Mach. v. United States, 201 F.3d 1367, 1372 (Fed. Cir. 2000). The plain meaning of

“component” is not limited to tangible items. “Component” is consistently defined as a “constituent element” or a “constituent part.” Webster’s Third New International Dictionary 466 (3d ed. 1986); The American Heritage Dictionary 387 (3d ed. 1992).

This Court has previously allowed § 271(f) claims involving software. Southwest Software, Inc. v. Harlequin Inc., 226 F.3d 1280, 1298-99 (Fed. Cir. 2000). There, this Court vacated the district court’s judgment of non-infringement of accused software products, and remanded the case for further proceedings on the infringement issue, including infringement under § 271(f). Id.

Furthermore, each district court that has considered the issue, including the court below, has concluded that there is no basis to exclude software from the definition of “component” in § 271(f). A000051-54; AT&T Corp. v. Microsoft Corp., No. 01 Civ. 4872, 2004 U.S. Dist. LEXIS 3340 (S.D.N.Y. Mar. 5, 2004); Imagexpo, L.L.C. v. Microsoft Corp., 299 F. Supp. 2d 550 (E.D. Va. 2003); NTP, Inc. v. Research in Motion, Ltd., 261 F. Supp. 2d 423, 431 (E.D. Va. 2002).

The PTO acknowledges that “software” is a “component” of a patentable invention. The MPEP, in discussing the patentability of computer-related inventions, consistently refers to “software *components*.” Manual of Patent Examining Procedure § 2106, 2100-13 (8th ed. 2003) (emphasis added); see also § 2106.01, 2100-24; § 2106.02, 2100-25.

Microsoft's reliance on dicta from Bayer AG v. Housey Pharms., Inc., 340 F.3d 1367 (Fed. Cir. 2003), which is not even a case about § 271(f), is misplaced. There, the patentee alleged that Bayer used the patented process outside the United States to generate certain information and imported that information to the United States, where it used its own processes to carry out the manufacture of the identified drugs. Id. at 1369-70. This Court held that Bayer's importation of the *information* was not importation of a "product" under § 271(g). Id. at 1377-78.

In the present case, there is simply no issue whether Microsoft's Internet Explorer software is a "product," as claimed in the '906 patent. Under § 271(f), unlike § 271(g), all that is required is that Microsoft's software be a "component" of that "patented invention." It certainly is, because that software – that set of instructions – becomes *part* of the programmed machines on which it is installed abroad. That is not only common sense but the essence of this Court's longstanding approach to the patentability of software as embodied in a physical medium. See In re Alappat, 33 F.3d 1526, 1545 (Fed. Cir. 1994) (en banc). "Information" as involved in Bayer, on the other hand, is not patentable at all. See Diamond v. Diehr, 450 U.S. 175, 185 (1981).

Microsoft's argument that software code is not "combined" with anything abroad, but is merely replicated fails for a host of reasons. First, the characterization

of the process as “replication” is not correct. The code is in fact “installed” on hardware abroad. A003094-95. That “installation,” admitted Microsoft, combines the code with the “computer usable medium” called for in claim 6 of the ’906 patent to make a “computer program product.” Id.

Installation falls within the definition of “combine.” Dictionaries define “combine” as “[t]o bring into a state of unity; merge” and “to bring into close relationship.” The American Heritage Dictionary 377 (3d ed. 1992); Webster’s Third New International Dictionary 452 (3d ed. 1986). There is no limitation in these definitions that requires a physical joining of tangible components.

The creation of a computer program product is a process that “combines” software and hardware. WMS Gaming, Inc. v. Int’l Game Tech., 184 F.3d 1339, 1348-49 (Fed. Cir. 1999) (general purpose computer becomes special purpose one once programmed); In re Alappat, 33 F.3d at 1545 (same). Thus, “a hardware and software *combination* [] defines a statutory product.” Manual of Patent Examining Procedure § 2106, 2100-14 (8th ed. 2003) (emphasis added).

For this reason, Johns Hopkins Univ. v. CellPro, Inc., 152 F.3d 1342 (Fed. Cir. 1998) is inapplicable. Johns Hopkins does not address § 271(f). Rather, it addresses whether shipments of stem cells to Canada prior to the issuance of a patent violates

United States patent laws. Id. at 1365-67. There, the cell lines were replicated, as opposed to the process of installation that occurs here.

Finally, Microsoft is wrong not just about whether § 271(f) applies but about what the effect of its inapplicability would be, contending that the consequence would be reduction of damages to \$161,374,307. Microsoft never presented any evidence on the proper division of domestic and export sales, although, as the district court and Microsoft's counsel understood, it was not precluded from doing so (damages for infringement of method claim 1 were limited to units made, used or sold in the United States under § 271(a)). A100953 (Microsoft's counsel confirming that: "[U]nits will be broken out as between the U.S. and foreign"). The only evidence on the allocation was offered by Plaintiffs-Appellees. A100966; A100979; A151550-79; A151663-64. That evidence requires, if this Court were to grant Microsoft's requested relief, a reduction of the damages award to \$299,597,946.69 (203,808,127 U.S. units multiplied by \$1.47). Id.; A000047-49.

CONCLUSION

For the foregoing reasons, Plaintiffs-Appellees respectfully request that this Court affirm the verdict and judgment below.

Respectfully submitted,

Martin R. Lueck/man

Martin R. Lueck

Jan M. Conlin

Richard M. Martinez

Munir R. Meghjee

ROBINS, KAPLAN, MILLER & CIRESI L.L.P.

2800 LaSalle Plaza

800 LaSalle Avenue

Minneapolis, MN 55402-2015

(612) 349-8500

ATTORNEYS FOR PLAINTIFFS-APPELLEES
EOLAS TECHNOLOGIES INCORPORATED AND
THE REGENTS OF THE UNIVERSITY
OF CALIFORNIA