

NLWJC - Kagan

Counsel - Box 032 - Folder 002

Lotus v. Borland [6]

Supreme Court, U.S.
FILED
No.

942003 JUN 7 - 1995

IN THE
OFFICE OF THE CLERK
Supreme Court of the United States
OCTOBER TERM, 1994

LOTUS DEVELOPMENT CORPORATION,

Petitioner,

—v.—

BORLAND INTERNATIONAL, INC.,

Respondent.

ON PETITION FOR A WRIT OF CERTIORARI TO THE UNITED STATES
COURT OF APPEALS FOR THE FIRST CIRCUIT

APPENDIX TO PETITION FOR A WRIT OF CERTIORARI

CHARLES FRIED
1575 Massachusetts Avenue
Cambridge, Massachusetts 02138
(617) 495-4636

ARTHUR R. MILLER
1545 Massachusetts Avenue
Cambridge, Massachusetts 02138
(617) 495-4111

HENRY B. GUTMAN
Counsel of Record
KERRY L. KONRAD
KAREN F. CONWAY
JEFFREY E. OSTROW
ANNE W. SALISBURY
BAKER & BOTTS, L.L.P.
885 Third Avenue
New York, New York 10022
(212) 705-5000

THOMAS M. LEMBERG
LOTUS DEVELOPMENT
CORPORATION
55 Cambridge Parkway
Cambridge, Massachusetts 02142
(617) 693-8106

Counsel for Petitioner

INDEX TO APPENDIX

	PAGE
Opinion of the United States Court of Appeals for the First Circuit in <i>Lotus Development Corporation v. Borland International, Inc.</i> , dated March 9, 1995.	1a
Opinion of the United States District Court for the District of Massachusetts in <i>Lotus Development Corporation v. Borland International, Inc.</i> , dated August 12, 1993, as amended August 19, 1993 ...	29a
Permanent Injunction entered by United States District Court of Massachusetts in <i>Lotus Development Corporation v. Borland International, Inc.</i> , dated August 19, 1993.....	69a
Opinion of the United States District Court for the District of Massachusetts in <i>Lotus Development Corporation v. Borland International, Inc.</i> , dated June 30, 1993, as amended August 19, 1993	71a
Opinion of the United States District Court for the District of Massachusetts in <i>Lotus Development Corporation v. Borland International, Inc.</i> , dated July 31, 1992.....	106a
Opinion of the United States District Court for the District of Massachusetts in <i>Lotus Development Corporation v. Borland International, Inc.</i> , dated March 20, 1992.....	145a
Opinion of the United States District Court for the District of Massachusetts in <i>Lotus Development Corporation v. Paperback Software International</i> , dated June 28, 1990	183a

	PAGE
The 1976 Copyright Act, 17 U.S.C. §§ 101, 102(a), 102(b), 103, 117	270a
Affidavit of Mitchell D. Kapur, dated November 27, 1989, submitted in <i>Lotus Development Corporation</i> <i>v. Borland International, Inc. and Lotus</i> <i>Development Corporation v. Paperback Software</i> <i>International</i>	281a

UNITED STATES COURT OF APPEALS
FOR THE FIRST CIRCUIT

No. 93-2214

LOTUS DEVELOPMENT CORPORATION,
Plaintiff, Appellee,

—v.—

BORLAND INTERNATIONAL, INC.,
Defendant, Appellant.

Appeal from the United States District Court
for the District of Massachusetts
[Hon. Robert E. Keeton *U.S. District Judge*]

Before
TORRUELLA, *Chief Judge,*
BOUDIN and STAHL, *Circuit Judges.*

Gary L. Reback, with whom Peter N. Detkin, Michael Barclay, Isabella E. Fu, Wilson Sonsini Goodrich & Rosati, Peter E. Gelhaar, Katherine L. Parks, and Donnelly Conroy & Gelhaar, were on brief for appellant.

Matthew P. Poppel, et al., were on brief for Computer Scientists, amicus curiae.

Dennis S. Karjala, and Peter S. Menell, on brief, amici curiae.

Jeffrey C. Cannon and Baker Keaton Seibel & Cannon, were on brief for Computer Software Industry Ass'n, amicus curiae.

Laureen E. McGurk, David A. Rabin, Bryan G. Harrison and Morris Manning & Martin, were on brief for Chicago Computer Soc., Diablo Users Group, Danbury Area Computer Soc., IBM AB Users Group, Kentucky-Indiana Personal Computer Users Group, Long Island PC Users Group, Napa Valley PC Users Group, Pacific Northwest PC Users Group, Palmetto Personal Computer Club, Philadelphia Area Computer Soc., Inc., Phoenix IBM PC Users Group, Pinellas IBM PC Users Group, Quad Cities Computer Soc., Quattro Pro Users Group, Sacramento PC Users Group, San Francisco PC Users Group, Santa Barbara PC Users Group, Twin Cities PC Users Group, and Warner Robbins Personal Computer Ass'n, amici curiae.

Diane Marie O'Malley and Hanson Bridgett Marcus Vlahos & Rudy, were on brief for Software Entrepreneurs' Forum, amicus curiae.

Peter M.C. Choy, was on brief for American Committee for Interoperable Systems, amicus curiae.

Howard B. Abrams, Howard C. Anawalt, Stephen R. Barnett, Ralph S. Brown, Stephen L. Carter, Amy B. Cohen, Paul J. Heald, Peter A. Jaszi, John A. Kidwell, Edmund W. Kitch, Roberta R. Kwall, David L. Lange, Marshall Leaffer, Jessica D. Litman, Charles R. McManis, L. Ray Patterson, Jerome H. Reichman, David A. Rice, Pamela Samuelson, David J. Seipp, David E. Shipley, Lionel S. Sobel, Alfred C. Yen, and Diane L. Zimmerman, were on brief for Copyright Law Professors, amicus curiae.

Henry B. Gutman, with whom *Kerry L. Konrad, Joshua H. Epstein, Kimberly A. Caldwell, O'Sullivan Graev & Karabell, Thomas M. Lemberg, James C. Burling, and Hale and Dorr*, were on brief for appellee.

Morton David Goldberg, June M. Besek, David O. Carson, Jesse M. Feder, Schwab Goldberg Price & Dannay, and

Arthur R. Miller, were on brief for Apple Computer, Inc., Digital Equip. Corp., International Business Machines Corp., and Xerox Corp., amici curiae.

Jon A. Baumgarten, Proskauer Rose Goetz & Mendelsohn, and Robert A. Gorman, were on brief for Adobe Systems, Inc., Apple Computer, Inc., Computer Associates Intern., Inc., Digital Equip. Corp., and International Business Machines Corp., amici curiae.

Herbert F. Schwartz, Vincent N. Palladino, Susan Prohoff, Fish & Neave, William J. Cheeseman, and Foley Hoag & Eliot, were on brief for Computer and Business Equip. Mfrs. Ass'n, amicus curiae.

STAHL, Circuit Judge.

This appeal requires us to decide whether a computer menu command hierarchy is copyrightable subject matter. In particular, we must decide whether, as the district court held, plaintiff-appellee Lotus Development Corporation's copyright in Lotus 1-2-3, a computer spreadsheet program, was infringed by defendant-appellant Borland International, Inc., when Borland copied the Lotus 1-2-3 menu command hierarchy into its Quattro and Quattro Pro computer spreadsheet programs. See *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 788 F.Supp. 78 (D.Mass. 1992) ("*Borland I*"); *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 799 F.Supp. 203 (D.Mass. 1992) ("*Borland II*"); *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 831 F.Supp. 202 (D.Mass. 1993) ("*Borland III*"); *Lotus Dev. Corp. v. Borland Int'l Inc.*, 831 F.Supp. 223 (D.Mass. 1993) ("*Borland IV*").

I.

Background

Lotus 1-2-3 is a spreadsheet program that enables users to perform accounting functions electronically on a computer. Users manipulate and control the program via a series of menu commands, such as "Copy," "Print," and "Quit." Users choose commands either by highlighting them on the screen or by typing their first letter. In all, Lotus 1-2-3 has 469 commands arranged into more than 50 menus and submenus.

Lotus 1-2-3, like many computer programs, allows users to write what are called "macros." By writing a macro, a user can designate a series of command choices with a single macro keystroke. Then, to execute that series of commands in multiple parts of the spreadsheet, rather than typing the whole series each time, the user only needs to type the single pre-programmed macro keystroke, causing the program to recall and perform the designated series of commands automatically. Thus, Lotus 1-2-3 macros shorten the time needed to set up and operate the program.

Borland released its first Quattro program to the public in 1987, after Borland's engineers had labored over its development for nearly three-years. Borland's objective was to develop a spreadsheet program far superior to existing programs, including Lotus 1-2-3. In Borland's words, "[f]rom the time of its initial release . . . Quattro included enormous innovations over competing spreadsheet products."

The district court found, and Borland does not now contest, that Borland included in its Quattro and Quattro Pro version 1.0 programs "a *virtually identical* copy of the entire 1-2-3 menu tree." *Borland III*, 831 F.Supp. at 212 (emphasis in original). In so doing, Borland did not copy any of Lotus's underlying computer code; it copied only the words and structure of Lotus's menu command hierarchy. Borland included the Lotus menu command hierarchy in its programs to make them compatible with Lotus 1-2-3 so that spreadsheet users who were already familiar with Lotus 1-2-3 would be able to

switch to the Borland programs without having to learn new commands or rewrite their Lotus macros.

In its Quattro and Quattro Pro version 1.0 programs, Borland achieved compatibility with Lotus 1-2-3 by offering its users an alternate user interface, the "~~Lotus Emulation Interface~~." By activating the Emulation Interface, Borland users would see the Lotus menu commands on their screens and could interact with Quattro or Quattro Pro as if using Lotus 1-2-3, albeit with a slightly different looking screen and with many Borland options not available on Lotus 1-2-3. In effect, Borland allowed users to choose how they wanted to communicate with Borland's spreadsheet programs: either by using menu commands designed by Borland, or by using the commands and command structure used in Lotus 1-2-3 augmented by Borland-added commands.

Lotus filed this action against Borland in the District of Massachusetts on July 2, 1990, four days after a district court held that the Lotus 1-2-3 "menu structure, taken as a whole—including the choice of command terms [and] the structure and order of those terms," was protected expression covered by Lotus's copyrights. *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F.Supp. 37, 68, 70 (D.Mass.1990) ("*Paperback*").¹ Three days earlier, on the morning after the *Paperback* decision, Borland had filed a declaratory judgment action against Lotus in the Northern District of California, seeking a declaration of non-infringement. On September 10, 1990, the district court in California dismissed Borland's declaratory judgment action in favor of this action.

Lotus and Borland filed cross motions for summary judgment; the district court denied both motions on March 20, 1992, concluding that "neither party's motion is supported by the record." *Borland I*, 788 F.Supp. at 80. The district court invited the parties to file renewed summary judgment motions that would "focus their arguments more precisely" in light of rulings it had made in conjunction with its denial of their summary judgment motions. *Id.* at 82. Both parties filed

¹ Judge Keeton presided over both the *Paperback* litigation and this case.

renewed motions for summary judgment on April 24, 1992. In its motion, Borland contended that the Lotus 1-2-3 menus were not copyrightable as a matter of law and that no reasonable trier of fact could find that the similarity between its products and Lotus 1-2-3 was sufficient to sustain a determination of infringement. Lotus contended in its motion that Borland had copied Lotus 1-2-3's entire user interface and had thereby infringed Lotus's copyrights.

On July 31, 1992, the district court denied Borland's motion and granted Lotus's motion in part. The district court ruled that the Lotus menu command hierarchy was copyrightable expression because

[a] very satisfactory spreadsheet menu tree can be constructed using different commands and a different command structure from those of Lotus 1-2-3. In fact, Borland has constructed just such an alternate tree for use in Quattro Pro's native mode. Even if one holds the arrangement of menu commands constant, it is possible to generate literally millions of satisfactory menu trees by varying the menu commands employed.

Borland II, 799 F.Supp. at 217. The district court demonstrated this by offering alternate command words for the ten commands that appear in Lotus's main menu. *Id.* For example, the district court stated that "[t]he 'Quit' command could be named 'Exit' without any other modifications," and that "[t]he 'Copy' command could be called 'Clone,' 'Ditto,' 'Duplicate,' 'Imitate,' 'Mimic,' 'Replicate,' and 'Reproduce,' among others." *Id.* Because so many variations were possible, the district court concluded that the Lotus developers' choice and arrangement of command terms, reflected in the Lotus menu command hierarchy, constituted copyrightable expression.

In granting partial summary judgment to Lotus, the district court held that Borland had infringed Lotus's copyright in Lotus 1-2-3:

[A]s a matter of law, Borland's Quattro products infringe the Lotus 1-2-3 copyright because of (1) the extent of

copying of the "menu commands" and "menu structure" that is not *genuinely* disputed in this case, (2) the extent to which the copied elements of the "menu commands" and "menu structure" contain expressive aspects separable from the functions of the "menu commands" and "menu structure," and (3) the scope of those copied expressive aspects as an integral part of Lotus 1-2-3.

Borland II, 799 F.Supp. at 223 (emphasis in original). The court nevertheless concluded that while the Quattro and Quattro Pro programs infringed Lotus's copyright, Borland had not copied the entire Lotus 1-2-3 user interface, as Lotus had contended. Accordingly, the court concluded that a jury trial was necessary to determine the scope of Borland's infringement, including whether Borland copied the long prompts² of Lotus 1-2-3, whether the long prompts contained expressive elements, and to what extent, in any, functional constraints limited the number of possible ways that the Lotus menu command hierarchy could have been arranged at the time of its creation. *See Borland III*, 831 F.Supp. at 207. Additionally, the district court granted Lotus summary judgment on Borland's affirmative defense of waiver, but not on its affirmative defenses of laches and estoppel. *Borland II*, 799 F.Supp. at 222-23.

² Lotus 1-2-3 utilizes a two-line menu; the top line lists the commands from which the user may choose, and the bottom line displays what Lotus calls its "long prompts." The long prompts explain, as a sort of "help text," what the highlighted menu command will do if entered. For example, the long prompt for the "Worksheet" command displays the submenu that the "Worksheet" command calls up; it reads "Global, Insert, Delete, Column, Erase, Titles, Window, Status, Page." The long prompt for the "Copy" command explains what function the "Copy" command will perform: "Copy a cell or range of cells." The long prompt for the "Quit" command reads, "End 1-2-3 session (Have you saved your work?)."

Prior to trial, the parties agreed to exclude the copying of the long prompts from the case; Lotus agreed not to contend that Borland had copied the long prompts, Borland agreed not to argue that it had not copied the long prompts, and both sides agreed not to argue that the issue of whether Borland had copied the long prompts was material to any other issue in the case. *See Borland III*, 831 F.Supp. at 208.

Immediately following the district court's summary judgment decision, Borland removed the Lotus Emulation Interface from its products. Thereafter, Borland's spreadsheet programs no longer displayed the Lotus 1-2-3 menus to Borland users, and as a result Borland users could no longer communicate with Borland's programs as if they were using a more sophisticated version of Lotus 1-2-3. Nonetheless, Borland's programs continued to be partially compatible with Lotus 1-2-3, for Borland retained what it called the "Key Reader" in its Quattro Pro programs. Once turned on, the Key Reader allowed Borland's programs to understand and perform some Lotus 1-2-3 macros.³ With the Key Reader on, the Borland programs used Quattro Pro menus for display, interaction, and macro execution, except when they encountered a slash ("/") key in a macro (the starting key for any Lotus 1-2-3 macro), in which case they interpreted the macro as having been written for Lotus 1-2-3. Accordingly, people who wrote or purchased macros to shorten the time needed to perform an operation in Lotus 1-2-3, could still use those macros in Borland's programs.⁴ The district court permitted Lotus to file a supplemental complaint alleging that the Key Reader infringed its copyright.

The parties agreed to try the remaining liability issues with-out a jury. The district court held two trials, the Phase I trial covering all remaining issues raised in the original complaint (relating to the Emulation Interface) and the Phase II trial covering all issues raised in the supplemental complaint (relating to the Key Reader). At the Phase I trial, there were no live witnesses, although considerable testimony was presented in the form of affidavits and deposition excerpts. The district court ruled upon evidentiary objections counsel interposed. At the Phase II trial, there were two live witnesses,

³ Because Borland's programs could no longer display the Lotus menu command hierarchy to users, the Key Reader did not allow debugging or modification of macros, nor did it permit the execution of most interactive macros.

⁴ See *Borland IV*, 831 F.Supp. at 226-27, for a more detailed explanation of macros and the Key Reader.

each of whom demonstrated the programs for the district court.

After the close of the Phase I trial, the district court permitted Borland to amend its answer to include the affirmative defense of "fair use." Because Borland had presented all of the evidence supporting its fair-use defense during the Phase I trial, but Lotus had not presented any evidence on fair use (as the defense had not been raised before the conclusion of the Phase I trial), the district court considered Lotus's motion for judgment on partial findings of fact. See Fed.R.Civ.P. 52(c). The district court held that Borland had failed to show that its use of the Lotus 1-2-3 menu command hierarchy in its Emulation Interface was a fair use. See *Borland III*, 831 F.Supp. at 208.

In its Phase I-trial decision, the district court found that "each of the Borland emulation interfaces contains a virtually identical copy of the 1-2-3 menu tree and that the 1-2-3 menu tree is capable of a wide variety of expression." *Borland III*, 831 F.Supp. at 218. The district court also rejected Borland's affirmative defenses of laches and estoppel. *Id.* at 218-23.

In its Phase II-trial decision, the district court found that Borland's Key Reader file included "a virtually identical copy of the Lotus menu tree structure, but represented in a different form and with first letters of menu command names in place of the full menu command names." *Borland IV*, 831 F.Supp. at 228. In other words, Borland's programs no longer included the Lotus command terms, but only their first letters. The district court held that "the Lotus menu structure, organization, and first letters of the command names . . . constitute part of the protectable expression found in [Lotus 1-2-3]." *Id.* at 233. Accordingly, the district court held that with its Key Reader, Borland had infringed Lotus's copyright. *Id.* at 245. The district court also rejected Borland's affirmative defenses of waiver, laches, estoppel, and fair use. *Id.* at 235-45. The district court then entered a permanent injunction against Borland, *id.* at 245, from which Borland appeals.

This appeal concerns only Borland's copying of the Lotus menu command hierarchy into its Quattro programs and Borland's affirmative defenses to such copying. Lotus has not

cross-appealed; in other words, Lotus does not contend on appeal that the district court erred in finding that Borland had not copied other elements of Lotus 1-2-3, such as its screen displays.

II.

Discussion

On appeal, Borland does not dispute that it factually copied the words and arrangement of the Lotus menu command hierarchy. Rather, Borland argues that it "lawfully copied the unprotectable menus of Lotus 1-2-3." Borland contends that the Lotus menu command hierarchy is not copyrightable because it is a system, method of operation, process, or procedure foreclosed from protection by 17 U.S.C. § 102(b). Borland also raises a number of affirmative defenses.

A. Copyright Infringement Generally

To establish copyright infringement, a plaintiff must prove "(1) ownership of a valid copyright, and (2) copying of constituent elements of the work that are original." *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 361, 111 S.Ct. 1282, 1296, 113 L.Ed.2d 358 (1991); see also *Data Gen. Corp. v. Grumman Sys. Support Corp.*, 36 F.3d 1147, 1160 n. 19 (1st Cir.1994); *Concrete Mach. Co. v. Classic Lawn Ornaments, Inc.*, 843 F.2d 600, 605 (1st Cir.1988). To show ownership of a valid copyright and therefore satisfy *Feist's* first prong, a plaintiff must prove that the work as a whole is original and that the plaintiff complied with applicable statutory formalities. See *Engineering Dynamics, Inc. v. Structural Software, Inc.*, 26 F.3d 1335, 1340 (5th Cir.1994). "In judicial proceedings, a certificate of copyright registration constitutes *prima facie* evidence of copyrightability and shifts the burden to the defendant to demonstrate why the copyright is not valid." *Bibbero Sys., Inc. v. Colwell Sys., Inc.*, 893 F.2d 1104, 1106 (9th Cir.1990); see also 17 U.S.C. § 410(c); *Folio Impressions, Inc. v. Byer California*, 937 F.2d 759, 763 (2d Cir.1991) (presumption of validity may be rebutted).

To show actionable copying and therefore satisfy *Feist's* second prong, a plaintiff must first prove that the alleged infringer copied plaintiff's copyrighted work as a factual matter; to do this, he or she may either present direct evidence of factual copying or, if that is unavailable, evidence that the alleged infringer had access to the copyrighted work and that the offending and copyrighted works are so similar that the court may infer that there was factual copying (i.e., probative similarity). *Engineering Dynamics*, 26 F.3d at 1340; see also *Concrete Mach.*, 843 F.2d at 606. The plaintiff must then prove that the copying of copyrighted material was so extensive that it rendered the offending and copyrighted works substantially similar. See *Engineering Dynamics*, 26 F.3d at 1341.

In this appeal, we are faced only with whether the Lotus menu command hierarchy is copyrightable subject matter in the first instance, for Borland concedes that Lotus has a valid copyright in Lotus 1-2-3 as a whole⁵ and admits to factually copying the Lotus menu command hierarchy. As a result, this appeal is in a very different posture from most copyright-infringement cases, for copyright infringement generally turns or whether the defendant has copied protected expression as a factual matter. Because of this different posture, most copyright-infringement cases provide only limited help to us in deciding this appeal. This is true even with respect to those copyright-infringement cases that deal with computers and computer software.

⁵ Computer programs receive copyright protection as "literary works." See 17 U.S.C. § 102(a)(1) (granting protection to "literary works") and 17 U.S.C. § 101 (defining "literary works" as "works . . . expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as books, periodicals, phonorecords, film, tapes, disks, or cards, in which they are embodied" (emphasis added)); see also H.R. Rep. No. 1476, 94th Cong., 2d Sess. 54 (1976), reprinted in 1976 U.S.C.C.A.N. 5659, 5667 ("The term 'literary works' . . . includes computer data bases, and computer programs to the extent that they incorporate authorship in the programmer's expression of original ideas, as distinguished from the ideas themselves.").

B. Matter of First Impression

Whether a computer menu command hierarchy constitutes copyrightable subject matter is a matter of first impression in this court. While some other courts appear to have touched on it briefly in dicta, *see, e.g., Autoskill, Inc. v. National Educ. Support Sys., Inc.*, 994 F.2d 1476, 1495 n. 23 (10th Cir.), *cert. denied*, ___ U.S. ___, 114 S.Ct. 307, 126 L.Ed.2d 254 (1993), we know of no cases that deal with the copyrightability of a menu command hierarchy standing on its own (i.e., without other elements of the user interface, such as screen displays, in issue). Thus we are navigating in uncharted waters.

Borland vigorously argues, however, that the Supreme Court charted our course more than 100 years ago when it decided *Baker v. Selden*, 101 U.S. 99, 25 L.Ed. 841 (1879). In *Baker v. Selden*, the Court held that Selden's copyright over the textbook in which he explained his new way to do accounting did not grant him a monopoly on the use of his accounting system.⁶ Borland argues:

The facts of *Baker v. Selden*, and even the arguments advanced by the parties in that case, are identical to those in this case. The only difference is that the "user interface" of Selden's system was implemented by pen and paper rather than by computer.

To demonstrate that *Baker v. Selden* and this appeal both involve accounting systems, Borland even supplied this court with a video that, with special effects, shows Selden's paper forms "melting" into a computer screen and transforming into Lotus 1-2-3.

We do not think that *Baker v. Selden* is nearly as analogous to this appeal as Borland claims. Of course, Lotus 1-2-3 is a computer spreadsheet, and as such its grid of horizontal rows and vertical columns certainly resembles an accounting ledger or any other paper spreadsheet. Those grids, however, are not at issue in this appeal for, unlike Selden, Lotus does not claim to have a monopoly over its accounting system. Rather, this

⁶ Selden's system of double-entry bookkeeping is the now almost-universal T-accounts system.

appeal involves Lotus's monopoly over the commands it uses to operate the computer. Accordingly, this appeal is not, as Borland contends, "identical" to *Baker v. Selden*.

C. *Altai*

Before we analyze whether the Lotus menu command hierarchy is a system, method of operation, process, or procedure, we first consider the applicability of the test the Second Circuit set forth in *Computer Assoc. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693 (2d Cir.1992).⁷ The Second Circuit designed its *Altai* test to deal with the fact that computer programs, copyrighted as "literary works," can be infringed by what is known as "nonliteral" copying, which is copying that is paraphrased or loosely paraphrased rather than word for word. *See id.* at 701 (citing nonliteral-copying cases); *see also* 3 Melville B. Nimmer & David Nimmer, *Nimmer on Copyright* § 13.03[A][1] (1993). When faced with nonliteral-copying cases, courts must determine whether similarities are due merely to the fact that the two works share the same underlying idea or whether they instead indicate that the second author copied the first author's expression. The Second Circuit designed its *Altai* test to deal with this situation in the computer context, specifically with whether one computer program copied nonliteral expression from another program's code.

The *Altai* test involves three steps: abstraction, filtration, and comparison. The abstraction step requires courts to "dissect the allegedly copied program's structure and isolate each level of abstraction contained within it." *Altai* 982 F.2d at 707. This step enables courts to identify the appropriate framework within which to separate protectable expression from unprotected ideas. Second, courts apply a "filtration"

⁷ We consider the *Altai* test because both parties and many of the *amici* focus on it so heavily. Borland, in particular, is highly critical of the district court for not employing the *Altai* test. Borland does not, however, indicate how using that test would have been dispositive in Borland's favor. Interestingly, Borland appears to contradict its own reasoning at times by criticizing the applicability of the *Altai* test.

step in which they examine "the structural components at each level of abstraction to determine whether their particular inclusion at that level was 'idea' or was dictated by considerations of efficiency, so as to be necessarily incidental to that idea; required by factors external to the program itself; or taken from the public domain." *Id.* Finally, courts compare the protected elements of the infringed work (i.e., those that survived the filtration screening) to the corresponding elements of the allegedly infringing work to determine whether there was sufficient copying of protected material to constitute infringement. *Id.* at 710.

In the instant appeal, we are not confronted with alleged nonliteral copying of computer code. Rather, we are faced with Borland's deliberate, literal copying of the Lotus menu command hierarchy. Thus, we must determine not whether nonliteral copying occurred in some amorphous sense, but rather whether the literal copying of the Lotus menu command hierarchy constitutes copyright infringement.

While the *Altai* test may provide a useful framework for assessing the alleged nonliteral copying of computer code, we find it to be of little help in assessing whether the literal copying of a menu command hierarchy constitutes copyright infringement. In fact, we think that the *Altai* test in this context may actually be misleading because, in instructing courts to abstract the various levels, it seems to encourage them to find a base level that includes copyrightable subject matter that, if literally copied, would make the copier liable for copyright infringement.⁸ While that base (or literal) level would not be at issue in a nonliteral-copying case like *Altai*, it is precisely what is at issue in this appeal. We think that abstracting menu command hierarchies down to their individual word and menu levels and then filtering idea from

⁸ We recognize that *Altai* never states that every work contains a copyrightable "nugget" of protectable expression. Nonetheless, the implication is that for literal copying, "it is not necessary to determine the level of abstraction at which similarity ceases to consist of an 'expression of ideas,' because literal similarity by definition is always a similarity as to the expression of ideas." 3 Melville B. Nimmer & David Nimmer, *Nimmer on Copyright* § 13.03[A](2) (1993).

expression at that stage, as both the *Altai* and the district court tests require, obscures the more fundamental question of whether a menu command hierarchy can be copyrighted at all. The initial inquiry should not be whether individual components of a menu command hierarchy are expressive, but rather whether the menu command hierarchy as a whole can be copyrighted. *But see Gates Rubber Co. v. Bando Chem. Indus., Ltd.*, 9 F.3d 823 (10th Cir.1993) (endorsing *Altai*'s abstraction-filtration-comparison test as a way of determining whether "menus and sorting criteria" are copyrightable).

D. *The Lotus Menu Command Hierarchy: A "Method of Operation"*

Borland argues that the Lotus menu command hierarchy is uncopyrightable because it is a system, method of operation, process, or procedure foreclosed from copyright protection by 17 U.S.C. § 102(b). Section 102(b) states: "In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work." Because we conclude that the Lotus menu command hierarchy is a method of operation, we do not consider whether it could also be a system, process, or procedure.

We think that "method of operation," as that term is used in § 102(b), refers to the means by which a person operates something, whether it be a car, a food processor, or a computer. Thus a text describing how to operate something would not extend copyright protection to the method of operation itself; other people would be free to employ that method and to describe it in their own words. Similarly, if a new method of operation is used rather than described, other people would still be free to employ or describe that method.

We hold that the Lotus menu command hierarchy is an uncopyrightable "method of operation." The Lotus menu command hierarchy provides the means by which users control and operate Lotus 1-2-3. If users wish to copy material, for example, they use the "Copy" command. If users wish to print material, they use the "Print" command. Users must use

of person

the command terms to tell the computer what to do. Without the menu command hierarchy, users would not be able to access and control, or indeed make use of, Lotus 1-2-3's functional capabilities.

The Lotus menu command hierarchy does not merely explain and present Lotus 1-2-3's functional capabilities to the user; it also serves as the method by which the program is operated and controlled. The Lotus menu command hierarchy is different from the Lotus long prompts, for the long prompts are not necessary to the operation of the program; users could operate Lotus 1-2-3 even if there were no long prompts.⁹ The Lotus menu command hierarchy is also different from the Lotus screen displays, for users need not "use" any expressive aspects of the screen displays in order to operate Lotus 1-2-3; because the way the screens look has little bearing on how users control the program, the screen displays are not part of Lotus 1-2-3's "method of operation."¹⁰ The Lotus menu command hierarchy is also different from the underlying computer code, because while code is necessary for the program to work, its precise formulation is not. In other words, to offer the same capabilities as Lotus 1-2-3, Borland did not have to copy Lotus's underlying code (and indeed it did not); to allow users to operate its programs in substantially the same way, however, Borland had to copy the Lotus menu command hierarchy. Thus the Lotus 1-2-3 code is not a uncopyrightable "method of operation."¹¹

⁹ As the Lotus long prompts are not before us appeal, we take no position on their copyrightability, although we do note that a strong argument could be made that the brief explanations they provide "merge" with the underlying idea of explaining such functions. See *Morrissey v. Procter & Gamble Co.*, 379 F.2d 675, 678-79 (1st Cir.1967) (when the possible ways to express an idea are limited, the expression "merges" with the idea and is therefore uncopyrightable; when merger occurs, identical copying is permitted).

¹⁰ As they are not before us on appeal, we take no position on whether the Lotus 1-2-3 screen displays constitute original expression capable of being copyrighted.

¹¹ Because the Lotus 1-2-3 code is not before us on appeal, we take no position on whether it is copyrightable. We note, however, that orig-

NOT A METHOD OF OPERATION SIC MORRISSEY FORMULATION NOT KEY TO NEW & REASON CREATES

The district court held that the Lotus menu command hierarchy, with its specific choice and arrangement of command terms, constituted an "expression" of the "idea" of operating a computer program with commands arranged hierarchically into menus and submenus. *Borland II*, 799 F.Supp. at 216. Under the district court's reasoning, Lotus's decision to employ hierarchically arranged command terms to operate its program could not foreclose its competitors from also employing hierarchically arranged command terms to operate their programs, but it did foreclose them from employing the specific command terms and arrangement that Lotus had used. In effect, the district court limited Lotus 1-2-3's "method of operation" to an abstraction.

Accepting the district court's finding that the Lotus developers made some expressive choices in choosing and arranging the Lotus command terms, we nonetheless hold that that expression is not copyrightable because it is part of Lotus 1-2-3's "method of operation." We do not think that "methods of operation" are limited to abstractions; rather, they are the means by which a user operates something. If specific words are essential to operating something, then they are part of a "method of operation" and, as such, are unprotectable. This is so whether they must be highlighted, typed in, or even spoken, as computer programs no doubt will soon be controlled by spoken words.

The fact that Lotus developers could have designed the Lotus menu command hierarchy differently is immaterial to the question of whether it is a "method of operation." In other words, our initial inquiry is not whether the Lotus menu command hierarchy incorporates any expression.¹² Rather, our initial inquiry is whether the Lotus menu command hierarchy is a "method of operation." Concluding, as we do, that users operate Lotus 1-2-3 by using the Lotus menu command hier-

inal computer codes generally are protected by copyright. See, e.g., *Altai*, 982 F.2d at 702 ("It is now well settled that the literal elements of computer programs, i.e., their source and object codes, are the subject of copyright protection.") (citing cases).

¹² We think that the *Altai* test would contemplate this being the initial inquiry.

archy, and that the entire Lotus menu command hierarchy is essential to operating Lotus 1-2-3, we do not inquire further whether that method of operation could have been designed differently. The "expressive" choices of what to name the command terms and how to arrange them do not magically change the uncopyrightable menu command hierarchy into copyrightable subject matter.

Our holding that "methods of operation" are not limited to mere abstractions is bolstered by *Baker v. Selden*. In *Baker*, the Supreme Court explained that

the teachings of science and the rules and methods of useful art have their final end in application and use; and this application and use are what the public derive from the publication of a book which teaches them. . . . The description of the art in a book, though entitled to the benefit of copyright, lays no foundation for an exclusive claim to the art itself. The object of the one is explanation; the object of the other is use. The former may be secured by copyright. The latter can only be secured, if it can be secured at all, by letters-patent.

Baker v. Selden, 101 U.S. at 104-05. Lotus wrote its menu command hierarchy so that people could learn it and use it. Accordingly, it falls squarely within the prohibition on copyright protection established in *Baker v. Selden* and codified by Congress in § 102(b).

In many ways, the Lotus menu command hierarchy is like the buttons used to control, say, a video cassette recorder ("VCR"). A VCR is a machine that enables one to watch and record video tapes. Users operate VCRs by pressing a series of buttons that are typically labeled "Record, Play, Reverse, Fast Forward, Pause, Stop/Eject." That the buttons are arranged and labeled does not make them a "literary work," nor does it make them an "expression" of the abstract "method of operating" a VCR via a set of labeled buttons. Instead, the buttons are themselves the "method of operating" the VCR.

When a Lotus 1-2-3 user chooses a command, either by highlighting it on the screen or by typing its first letter, he or

she effectively pushes a button. Highlighting the "Print" command on the screen, or typing the letter "P," is analogous to pressing a VCR button labeled "Play."

Just as one could not operate a buttonless VCR, it would be impossible to operate Lotus 1-2-3 without employing its menu command hierarchy. Thus the Lotus command terms are not equivalent to the labels on the VCR's buttons, but are instead equivalent to the buttons themselves. Unlike the labels on a VCR's buttons, which merely make operating a VCR easier by indicating the buttons' functions, the Lotus menu commands are essential to operating Lotus 1-2-3. Without the menu commands, there would be no way to "push" the Lotus buttons, as one could push unlabeled VCR buttons. While Lotus could probably have designed a user interface for which the command terms were mere labels, it did not do so here. Lotus 1-2-3 depends for its operation on use of the precise command terms that make up the Lotus menu command hierarchy.

One might argue that the buttons for operating a VCR are not analogous to the command for operating a computer program because VCRs are not copyrightable, where as computer programs are. VCRs may not be copyrighted because they do not fit within any of the § 102(a) categories of copyrightable works; the closest they come is "sculptural work." Sculptural works, however, are subject to a "useful-article" exception whereby "the design of a useful article . . . shall be considered a pictorial, graphic, or sculptural work only if, and only to the extent that, such design incorporates pictorial, graphic, or sculptural features that can be identified separately from, and are capable of existing independently of, the utilitarian aspects of the article." 17 U.S.C. § 101. A "useful article" is "an article having an intrinsic utilitarian function that is not merely to portray the appearance of the article or to convey information." *Id.* Whatever expression there may be in the arrangement of the parts of a VCR is not capable of existing separately from the VCR itself, so an ordinary VCR would not be copyrightable.

Computer programs, unlike VCRs, are copyrightable as "literary works." 17 U.S.C. § 102(a). Accordingly, one might

? why?

argue, the "buttons" used to operate a computer program are not like the buttons used to operate a VCR, for they are not subject to a useful-article exception. The response, of course, is that the arrangement of buttons on a VCR would not be copyrightable even without a useful-article exception, because the buttons are an uncopyrightable "method of operation." Similarly, the "buttons" of a computer program are also an uncopyrightable "method of operation."

That the Lotus menu command hierarchy is a "method of operation" becomes clearer when one considers program compatibility. Under Lotus's theory, if a user uses several different programs, he or she must learn how to perform the same operation in a different way for each program used. For example, if the user wanted the computer to print material, then the user would have to learn not just one method of operating the computer such that it prints, but many different methods. We find this absurd. The fact that there may be many different ways to operate a computer program, or even many different ways to operate a computer program using a set of hierarchically arranged command terms, does not make the actual method of operation chosen copyrightable; it still functions as a method for operating the computer and as such is uncopyrightable.

Consider also that users employ the Lotus menu command hierarchy in writing macros. Under the district court's holding, if the user wrote a macro to shorten the time needed to perform a certain operation in Lotus 1-2-3, the user would be unable to use that macro to shorten the time needed to perform that same operation in another program. Rather, the user would have to rewrite his or her macro using that other program's menu command hierarchy. This is despite the fact that the macro is clearly the user's own work product. We think that forcing the user to cause the computer to perform the same operation in a different way ignores Congress's direction in § 102(b) that "methods of operation" are not copyrightable. That programs can offer users the ability to write macros in many different ways does not change the fact that, once written, the macro allows the user to perform an operation automatically. As the Lotus menu command hierarchy

serves as the basis for Lotus 1-2-3 macros, the Lotus menu command hierarchy is a "method of operation."

In holding that expression that is part of a "method of operation" cannot be copyrighted, we do not understand ourselves to go against the Supreme Court's holding in *Feist*. In *Feist*, the Court explained:

The primary objective of copyright is not to reward the labor of authors, but to promote the Progress of Science and useful Arts. To this end, copyright assures authors the right to their original expression, but encourages others to build freely upon the ideas and information conveyed by a work.

Feist, 499 U.S. at 349-50, 111 S.Ct. at 1290 (quotations and citations omitted). We do not think that the Court's statement that "copyright assures authors the right to their original expression" indicates that all expression is necessarily copyrightable; while original expression is necessary for copyright protection, we do not think that it is alone sufficient. Courts must still inquire whether original expression falls within one of the categories foreclosed from copyright protection by § 102(b), such as being a "method of operation."

We also note that in most contexts, there is no need to "build" upon other people's expression, for the ideas conveyed by that expression can be conveyed by someone else without copying the first author's expression.¹³ In the context of methods of operation, however, "building" requires the use of the precise method of operation already employed; otherwise, "building" would require dismantling, too. Original developers are not the only people entitled to build on the methods of operation they create; anyone can. Thus, Borland may build on the method of operation that Lotus designed and may use the Lotus menu command hierarchy in doing so.

Our holding that methods of operation are not limited to abstractions goes against *Autoskill*, 994 F.2d at 1495 n. 23, in which the Tenth Circuit rejected the defendant's argument that

¹³ When there are a limited number of ways to express an idea, however, the expression "merges" with the idea and becomes uncopyrightable. *Morrissey*, 379 F.2d at 678-79.

the keying procedure used in a computer program was an uncopyrightable "procedure" or "method of operation" under § 102(b). The program at issue, which was designed to test and train students with reading deficiencies, *id.* at 1481, required students to select responses to the program's queries "by pressing the 1, 2, or 3 keys." *Id.* at 1495 n. 23. The Tenth Circuit held that, "for purposes of the preliminary injunction, . . . the record showed that [this] keying procedure reflected at least a minimal degree of creativity," as required by *Feist* for copyright protection. *Id.* As an initial matter, we question whether a programmer's decision to have users select a response by pressing the 1, 2, or 3 keys is original. More importantly, however, we fail to see how "a student select[ing] a response by pressing the 1, 2, or 3 keys," *id.*, can be anything but an unprotectable method of operation.¹⁴

III.

Conclusion

Because we hold that the Lotus menu command hierarchy is uncopyrightable subject matter, we further hold that Borland did not infringe Lotus's copyright by copying it. Accordingly, we need not consider any of Borland's affirmative defenses. The judgment of the district court is

Reversed.

Concurrence follows.

¹⁴ The Ninth Circuit has also indicated in dicta that "menus, and keystrokes" may be copyrightable. *Brown Bag Software v. Symantec Corp.*, 960 F.2d 1465, 1477 (9th Cir.), *cert. denied*, *BB Asset Management, Inc. v. Symantec Corp.*, ___ U.S. ___, 113 S.Ct. 198, 121 L.Ed.2d 141 (1992). In that case, however, the plaintiff did not show that the defendant had copied the plaintiff's menus or keystrokes, so the court was not directly faced with whether the menus or keystrokes constituted an unprotectable method of operation. *Id.*

BOUDIN, *Circuit Judge*, concurring.

The importance of this case, and a slightly different emphasis in my view of the underlying problem, prompt me to add a few words to the majority's tightly focused discussion.

I.

Most of the law of copyright and the "tools" of analysis have developed in the context of literary works such as novels, plays, and films. In this milieu, the principal problem—simply stated, if difficult to resolve—is to stimulate creative expression without unduly limiting access by others to the broader themes and concepts deployed by the author. The middle of the spectrum presents close cases; but a "mistake" in providing too much protection involves a small cost: subsequent authors treating the same themes must take a few more steps away from the original expression.

The problem presented by computer programs is fundamentally different in one respect. The computer program is a means for causing something to happen; it has a mechanical utility, an instrumental role, in accomplishing the world's work. Granting protection, in other words, can have some of the consequences of *patent* protection in limiting other people's ability to perform a task in the most efficient manner. Utility does not bar copyright (dictionaries may be copyrighted), but it alters the calculus.

Of course, the argument *for* protection is undiminished, perhaps even enhanced, by utility: if we want more of an intellectual product, a temporary monopoly for the creator provides incentives for others to create other, different items in this class. But the "cost" side of the equation may be different where one places a very high value on public access to a useful innovation that may be the most efficient means of performing a given task. Thus, the argument for extending protection may be the same; but the stakes on the other side are much higher.

It is no accident that patent protection has preconditions that copyright protection does not—notably, the requirements

of novelty and non-obviousness—and that patents are granted for a shorter period than copyrights. This problem of utility has sometimes manifested itself in copyright cases, such as *Baker v. Selden*, 101 U.S. 99, 25 L.Ed. 841 (1879), and been dealt with through various formulations that limit copyright or create limited rights to copy. But the case law and doctrine addressed to utility in copyright have been brief detours in the general march of copyright law.

Requests for the protection of computer menus present the concern with fencing off access to the commons in an acute form. A new menu may be a creative work, but over time its importance may come to reside more in the investment that has been made by users in learning the menu and in building their own mini-programs—macros—in reliance upon the menu. Better typewriter keyboard layouts may exist, but the familiar QWERTY keyboard dominates the market because that is what everyone has learned to use. See P. David, *CLIO and the Economics of QWERTY*, 75 *Am.Econ.Rev.* 332 (1985). The QWERTY keyboard is nothing other than a menu of letters.

Thus, to assume that computer programs are just one more new means of expression, like a filmed play, may be quite wrong. The “form”—the written source code or the menu structure depicted on the screen—look hauntingly like the familiar stuff of copyright; but the “substance” probably has more to do with problems presented in patent law or, as already noted, in those rare cases where copyright law has confronted industrially useful expressions. Applying copyright law to computer programs is like assembling a jigsaw puzzle whose pieces do not quite fit.

All of this would make no difference if Congress had squarely confronted the issue, and given explicit directions as to what should be done. The Copyright Act of 1976 took a different course. While Congress said that computer programs might be subject to copyright protection, it said this in very general terms; and, especially in § 102(b), Congress adopted a string of exclusions that if taken literally might easily seem to exclude most computer programs from protection. The only

detailed prescriptions for computers involve narrow issues (like back-up copies) of no relevance here.

Of course, one could still read the statute as a congressional command that the familiar doctrines of copyright law be taken and applied to computer programs, in cookie cutter fashion, as if the programs were novels or play scripts. Some of the cases involving computer programs embody this approach. It seems to be mistaken on two different grounds: the tradition of copyright law, and the likely intent of Congress.

The broad-brush conception of copyright protection, the time limits, and the formalities have long been prescribed by statute. But the heart of copyright doctrine—what may be protected and with what limitations and exceptions—has been developed by the courts through experience with individual cases. B. Kaplan, *An Unhurried View of Copyright* 40 (1967). Occasionally Congress addresses a problem in detail. For the most part the interstitial development of copyright through the courts is our tradition.

Nothing in the language or legislative history of the 1976 Act, or at least nothing brought to our attention, suggests that Congress meant the courts to abandon this case-by-case approach. Indeed, by setting up § 102(b) as a counterpoint theme, Congress has arguably recognized the tension and left it for the courts to resolve through the development of case law. And case law development is *adaptive*: it allows new problems to be solved with help of earlier doctrine, but it does not preclude new doctrines to meet new situations.

II.

In this case, the raw facts are mostly, if not entirely, undisputed. Although the inferences to be drawn may be more debatable, it is very hard to see that Borland has shown any interest in the Lotus menu except as a fall-back option for those users already committed to it by prior experience or in order to run their own macros using 1-2-3 commands. At least for the amateur, accessing the Lotus menu in the Borland Quattro or Quattro Pro program takes some effort.

Put differently, it is unlikely that users who value the Lotus menu for its own sake—independent of any investment they have made themselves in learning Lotus' commands or creating macros dependent upon them—would choose the Borland program in order to secure access to the Lotus menu. Borland's success is due primarily to other features. Its rationale for deploying the Lotus menu bears the ring of truth.

Now, any use of the Lotus menu by Borland is a commercial use and deprives Lotus of a portion of its "reward," in the sense that an infringement claim if allowed would increase Lotus' profits. But this is circular reasoning: broadly speaking, every limitation on copyright or privileged use diminishes the reward of the original creator. Yet not every writing is copyrightable or every use an infringement. The provision of reward is one concern of copyright law, but it is not the only one. If it were, copyrights would be perpetual and there would be no exceptions.

The present case is an unattractive one for copyright protection of the menu. The menu commands (e.g., "print," "quit") are largely for standard procedures that Lotus did not invent and are common words that Lotus cannot monopolize. What is left is the particular combination and sub-grouping of commands in a pattern devised by Lotus. This arrangement may have a more appealing logic and ease of use than some other configurations; but there is a certain arbitrariness to many of the choices.

If Lotus is granted a monopoly on this pattern, users who have learned the command structure of Lotus 1-2-3 or devised their own macros are locked into Lotus, just as a typist who has learned the QWERTY keyboard would be the captive of anyone who had a monopoly on the production of such a keyboard. Apparently, for a period Lotus 1-2-3 has had such sway in the market that it has represented the *de facto* standard for electronic spreadsheet commands. So long as Lotus is the superior spreadsheet—either in quality or in price—there may be nothing wrong with this advantage.

But if a better spreadsheet comes along, it is hard to see why customers who have learned the Lotus menu and devised macros for it should remain captives of Lotus because of an

investment in learning made by the users and not by Lotus. Lotus has already reaped a substantial reward for being first; assuming that the Borland program is now better, good reasons exist for freeing it to attract old Lotus customers: to enable the old customers to take advantage of a new advance, and to reward Borland in turn for making a better product. If Borland has not made a better product, then customers will remain with Lotus anyway.

Thus, for me the question is not whether Borland should prevail but on what basis. Various avenues might be traveled, but the main choices are between holding that the menu is not protectable by copyright and devising a new doctrine that Borland's use is privileged. No solution is perfect and no intermediate appellate court can make the final choice.

To call the menu a "method of operation" is, in the common use of those words, a defensible position. After all, the purpose of the menu is not to be admired as a work of literary or pictorial art. It is to transmit directions from the user to the computer, i.e., to operate the computer. The menu is also a "method" in the dictionary sense because it is a "planned way of doing something," an "order or system," and (aptly here) an "orderly or systematic arrangement, sequence or the like." *Random House Webster's College Dictionary* 853 (1991).

A different approach would be to say that Borland's use is privileged because, in the context already described, it is not seeking to appropriate the advances made by Lotus' menu; rather, having provided an arguably more attractive menu of its own, Borland is merely trying to give former Lotus users an option to exploit their own prior investment in learning or in macros. The difference is that such a privileged use approach would not automatically protect Borland if it had simply copied the Lotus menu (using different codes), contributed nothing of its own, and resold Lotus under the Borland label.

The closest analogue in conventional copyright is the fair use doctrine. E.g., *Harper & Row, Publishers, Inc. v. Nation Enters.*, 471 U.S. 539, 105 S.Ct. 2218, 85 L.Ed.2d 588 (1985). Although invoked by Borland, it has largely been brushed aside in this case because the Supreme Court has said

Policy
Are
Seems
Central

that it is "presumptively" unavailable where the use is a "commercial" one. See *id.* at 562, 105 S.Ct. at 2231-32. But see *Campbell v. Acuff-Rose Music, Inc.*, ___ U.S. ___, ___, 114 S.Ct. 1164, 1174, 127 L.Ed.2d 500 (1994). In my view, this is something less than a definitive answer; "presumptively" does not mean "always" and, in any event, the doctrine of fair use was created by the courts and can be adapted to new purposes.

But a privileged use doctrine would certainly involve problems of its own. It might more closely tailor the limits on copyright protection to the reasons for limiting that protection; but it would entail a host of administrative problems that would cause cost and delay, and would also reduce the ability of the industry to predict outcomes. Indeed, to the extent that Lotus' menu is an important standard in the industry, it might be argued that any use ought to be deemed privileged.

In sum, the majority's result persuades me and its formulation is as good, if not better, than any other that occurs to me now as within the reach of courts. Some solutions (*e.g.*, a very short copyright period for menus) are not options at all for courts but might be for Congress. In all events, the choices are important ones of policy, not linguistics, and they should be made with the underlying considerations in view.

UNITED STATES DISTRICT COURT
D. MASSACHUSETTS

Civ. A. No. 90-11662-K

Aug. 12, 1993.

As Amended Aug. 19, 1993.

Permanent Injunction Aug. 19, 1993.

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—v.—

BORLAND INTERNATIONAL, INC.,

Defendant.

James C. Burling, Jeffrey B. Rudman, Hale & Dorr, Boston, MA, Henry B. Gutman, Kerry L. Konrad, O'Sullivan, Graev & Karabell, New York City, *for plaintiff.*

Laura Steinberg, Sullivan & Worcester, Boston, MA, Lynn H. Pasahow, McCutchen, Doyle, Brown & Enersen, San Francisco, CA, David L. Hayes, Mitchell Zimmerman, Fenwick & West, Palo Alto, CA, Peter E. Gelhaar, Donnelly, Conroy & Gelhaar, Boston, MA, Gary L. Reback, Nina F. Locker, Peter N. Detkin, David A. Priebe, James A. DiBoise, Wilson, Soncini, Goodrich & Rosati, Palo Alto, CA, Laurence H. Tribe, Cambridge, MA, *for defendant.*

KEETON, District Judge.

A nonjury trial was held in this action in two phases. Phase I was tried on February 1-3, 1993; Phase II, on March 31-April 2, 1993. An Opinion of June 30, 1993 (the "Phase I Opinion") stated the court's findings and conclusions on issues raised in Phase I of the trial. The present Opinion states my findings and conclusions for the Phase II trial.

I. Introduction.

This Opinion assumes the reader's familiarity with, and follows the terminology set forth in, the Phase I Opinion. Background information appears also in *Lotus Dev. Corp. v. Borland Int'l Inc.*, 799 F.Supp. 203 (D.Mass.1992) ("*Borland II*"), *Lotus Dev. Corp. v. Borland Int'l Inc.*, 788 F.Supp. 78 (D.Mass.1992) ("*Borland I*"), and *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F.Supp. 37 (D.Mass.1990) ("*Paperback*"). —

As explained in the Phase I Opinion, the Phase I trial concerned issues of infringement raised in Lotus's original complaint, *ie.*, infringement by Borland's copying of Lotus's menu tree into the 1-2-3 emulation interface of Borland's Quattro and certain Quattro Pro spreadsheet programs.

Phase II of the trial concerns the full range of liability issues, including affirmative defenses, with respect to Lotus's supplemental complaint. The supplemental complaint alleges that Borland's "Key Reader" infringes Lotus's copyrights in its 1-2-3 programs.

II. Does Key Reader Infringe?

A. What is Key Reader?

The Phase I Opinion and the court's earlier Memorandum and Orders have focused on the 1-2-3 emulation interfaces in the Quattro and Quattro Pro programs. After executing the command that invokes the emulation interface in one of Borland's products, the program uses copies of the Lotus menu

tree for display to the user, interaction with the user (*i.e.*, interpreting the keystrokes typed by the user), and running macros.

The Key Reader is not a part of the emulation interface. The Key Reader feature may be turned on while the user continues to use another (*e.g.*, the Quattro) menu tree. When Key Reader has been turned on, the program continues to behave as it had before, with one exception. With Key Reader on, when the program encounters a slash key ("/") in a macro, the program interprets everything that follows the slash key as though it were part of a macro written for use with Lotus 1-2-3. Thus with Key Reader on, the program uses Quattro Pro menus for display, interaction and macro execution, except when a "/" key is encountered in a macro. Then, the program runs macros as though they were written using the 1-2-3 menu tree.

A more detailed explanation of 1-2-3 "macros" is essential to explaining Key Reader.

Generally, a 1-2-3 macro consists of some text contained in a spreadsheet cell. When the macro is invoked, the program begins at the left end of the text and treats each character in the text as though the user had typed that character into the 1-2-3 program. For example, consider a macro that consists of the text "/wp". When the macro is invoked, the program behaves precisely as though the user had typed into the regular interface a "/" (calling up the first menu) followed by "w" (selecting the "Worksheet" branch of the menu tree) followed by "p" (selecting the "Page" leaf; this executable operation inserts a page break into the spreadsheet).

In writing a macro, the user may use special commands not found in the 1-2-3 menu tree. For example, the character "" in a macro is interpreted as though the user struck the Enter key on the keyboard. Consider a macro consisting of the text "/r/c"". When invoked, this macro has the same effect as the user's typing into the 1-2-3 interface "/", "r", "f", "c", followed by striking the Enter key twice. That is, the program follows the "Range" branch, then the "Format" branch, then selects the "Currency" leaf. When the "Currency" leaf has

been selected, the program asks the user to specify the number of decimal places and the block of cells whose appearance is to be altered to that of monetary units. The first "" is equivalent to striking the Enter key; the program accepts the default number of decimal places (two). The program then asks for the range of cells. The second "" is equivalent to striking the Enter key a second time; the program accepts the default range (one cell).

Other sequences of symbols in a Lotus macro have a special meaning. Of particular interest is "{?}". When the program encounters this sequence of characters in a macro, the program pauses and waits for the user to strike keys appropriate for the menu tree. Thus, consider a Lotus macro consisting of the text "/rf{?}"". When invoked, this macro would follow the "r" command ("Range" from the first level of the Lotus menu tree), then the "f" command ("Format" in the Lotus menu tree). The program then encounters the "{?}" characters. Here, the program pauses and allows the user to input a menu-command. The program will interpret what the user inputs as a choice from the Lotus 1-2-3 submenu corresponding to "Range" "Format". Thus, the user might strike the "c" key (followed by Enter) to format a cell as currency. If the user strikes the Escape key instead of striking "c", the program backs up the menu tree from the "Format" submenu to the "Range" submenu. The user may continue maneuvering up and down the Lotus 1-2-3 menu tree until striking the Enter key. When the user strikes the Enter key, the program returns to reading the text of the macro as though the user were continuing by striking the keys found in the remaining text of the macro. A macro consisting only of "/{?}" permits a user to select any executable operation by typing keys precisely as though the user were using one of Borland's emulation interfaces, but without display of the menus.

Lotus 1-2-3 macros may also contain more advanced commands. Thus, a macro may contain text such as "{if . . .}" or "{let . . .}". In these strings of characters, the ellipses refer to additional text or "arguments" the user would provide. When the program encounters an advanced command such as

this, the program follows whatever procedures that command calls for. For example, the text "{(blank A1..G45)}" would cause the program to erase each of the cells from "A1" (first column and first row) through "G45" (seventh column and forty-fifth row). "if", "let", and "blank" are all labels in the Lotus 1-2-3 macro language that do not appear in the 1-2-3 menu tree.

In sum, when Key Reader is on and a slash key is encountered in a macro, the program follows the text of the macro as though the characters were being typed during the program's use of a copy of the 1-2-3 menu tree (and with some other means for interpreting the various special Lotus 1-2-3 macro language commands). When Key Reader is off, or no "/" key is encountered in a macro, the program interprets macros by reference to a different (*e.g.*, the Quattro) menu tree. With Key Reader off, Borland's programs cannot correctly interpret 1-2-3 macros.

Quattro and Quattro Pro version 1.0 did not contain the Key Reader feature; they contained only the emulation interface. Thus in these programs, the user could execute Lotus macros only when the user was employing the emulation interface; *ie.*, using Lotus 1-2-3 menus for display, interpreting user commands, and executing macros.

The Key Reader was first introduced in Quattro Pro version 2.0. Quattro Pro version 2.0, 3.0, and 4.0 contain *both* the emulation interface and the Key Reader feature (which is used from the native menus).

Borland removed the emulation interface from Quattro Pro version 4.01 (and subsequent releases of Quattro Pro) after the *Borland II* decision in this case allowed partial summary judgment for Lotus. Thus, Quattro Pro versions 4.01, SE, and Quattro Pro for Windows contain the Key Reader feature but have no 1-2-3 emulation interface.

B. Copying.

In developing Key Reader, Borland modified portions of the earlier Quattro Pro programs that contained the full 1-2-3

emulation interface. Accordingly, I begin with a review of how the Lotus menu tree was copied into the Quattro Pro emulation interface before analyzing how that was, in turn, incorporated into Key Reader.

1. The 123.MU file.

The actual menu tree for the Quattro Pro emulation interfaces (versions 1.0, 2.0, 3.0, and 4.0) was in a file labeled "123.MU". Thus, when the program was running and the user had selected the 1-2-3 emulation interface, the program would refer to the 123.MU file to determine the form of menu commands and menu structure used to present the set of executable operations to the user, interpret user commands, and interpret macros. If the user had selected a different interface, *e.g.*, one using the Quattro native menus, the program would refer to a different file for the menu commands and menu structure for display, interpretation of commands, and macro execution.

A text print-out of portions of the 123.MU file is in evidence under seal as Exhibit 13. The parties have included, however, the first page of this print-out in the public record. *See* Docket 311, Exh. A. An examination of the print-out demonstrates that the entire Lotus menu tree is copied into the file, with differences in indentation detailing the menu structure. That is, the structure of the menus and submenus is recorded in the file by changes in indentation. As explained in the Opinion for Phase I, when the program is using the 123.MU file as the source for its menus, it presents to the user a virtually identical copy of the Lotus menu tree (with Borland's additional menu commands inserted into the tree).

2. Key Reader files.

As the above explanation of macros suggests, interpretation of Lotus macros requires that the program treat the characters in the macro text as though they were keystrokes into a spreadsheet program using the Lotus menu tree (except for the characters in the macro text that are special commands in the Lotus macro language).

I find that, to implement the Key Reader (or at least that portion of the Key Reader that interprets characters from the Lotus menu tree), Borland began with the 123.MU file from the emulation interface. Borland then prepared a new file by reproducing the old 123.MU file but with only the first letter of each menu command name where the entire Lotus menu command name appears in the old 123.MU file. *See* Warfield Dep'n. IX:43. Put another way, the point is that to implement Key Reader Borland used a program file containing the same copy of the 1-2-3 menu tree structure and commands that Borland had used in its emulation interface, but with each menu command name stripped of everything after the first letter. Borland then appended this copy of the "stripped menu tree" to its quattro.mu file.

At various stages, Borland introduced other changes in how the menu structure is recorded in the stripped file, altering parentheses or indentation, etc. For example, I find that Borland may have altered the symbols used to record Lotus's menu structure, inserted the word "PICKLETTER" before the first letter of the menu command, or made other changes. *Compare* Exhibit 13 with Exhibit 517; *see* Warfield Dep'n, IX:43 at 148-49. Some of these things may actually appear in the old 123.MU file—Exhibit 13 contains only portions of the old 123.MU file. *See* Warfield Dep'n, IX:43 at 107-109, 148-49. In any event, I have considered the file and possible changes. I find that none of these changes is material to the scope or nature of copying from the Lotus 1-2-3 program.

In short, I find that the Key Reader file contains a virtually identical copy of the Lotus menu tree structure, but represented in a different form and with first letters of menu command names in place of the full menu command names.

Borland contends that the command letters copied from Lotus 1-2-3 that are in Borland's Key Reader files are not in the same order as displayed on the screen in Lotus 1-2-3. That is, if one reads sequentially down the Key Reader file, one encounters the Lotus commands in a different order than when reading the menus from the display screen while running Lotus 1-2-3. This is true, however, only in the sense that

the menu structure of Lotus 1-2-3 is represented in a different way in the Key Reader file than on the Lotus 1-2-3 display screens; in the file, the structure is detailed by differences in indentation (or other means) rather than through display on the screen. I find that the file (in evidence under seal, Exhibit 517) fully delineates a virtually identical copy of the menu structure of Lotus 1-2-3 including the first letter of each menu command in the corresponding location in the copy of the menu structure.

Because these menus are used in interpreting macros but are never fully displayed to the user, the parties have sometimes referred to these as "phantom menus."

In sum, to interpret macros, Borland's programs use a file with phantom menus consisting of a virtually identical copy of the Lotus menu tree that Borland used for its emulation interface, but with only the first letter of each menu command name where the complete menu command name previously appeared.

C. Copyrightability Issues.

1. Renewed and new arguments concerning definition of the idea.

Borland contends that copyright protection covering the structure of the 1-2-3 menu tree and the first letters of the commands in the 1-2-3 menu tree (*ie.*, Borland's phantom menus) would be equivalent to copyright protection for a "system" or "method" of communication between the user and the program. Thus, Borland argues that copying of the 1-2-3 menu tree structure and first letters of command names is a necessary part of any *system* for interpreting Lotus 1-2-3 macros. Citing the proposition that copyright law does not protect a "system", *see* 17 U.S.C. § 102(b), Borland argues that the Lotus copyright cannot extend to Borland's phantom menus.

Lotus responds that Borland has not proved that copying of any part of the menu tree is necessary for running or translating macros.

As a preliminary matter, I observe that the parties, witnesses (and the court, in some instances) have not used the terms "macro translation," "macro conversion," "macro execution," and "macro compatibility" with precision during the Phase II trial and earlier proceedings. In order to avoid possible future misunderstanding, I will explain the ambiguity in use of these terms before proceeding farther.

Macro "conversion" refers to translating a macro that was written using Lotus 1-2-3 into a macro written in a form for use in another program. For example, the macro "/rfc" in Lotus may be translated, using the macro translation assistant in Excel 2.1, into an Excel 2.1 macro having the text:

```
= SET.NAME("Selection—Save", SELECTION( ))
= SET.NAME("Range3", SELECTION( ))
= DISPLAY(FALSE)
= SELECT(Range3)
= FORMAT.NUMBER
("$#,##0.00;($#,##0.00)")
= RETURN( )
```

This macro text is written in the Excel macro language and is executed by reference to the Excel menus. Thus, formatting a single cell to be displayed in monetary units with two decimal places (the result of the "/rfc" macro in Lotus) may be achieved in Excel 2.1 by selecting a single cell in an Excel spreadsheet followed by selection of the "Format" menu command, the "Number" command in the submenu of Format, then the "\$#,##0.00;(\$#,##0.00)" (currency) leaf (compare with the second to last line of the Excel macro).

This facility for conversion of macros may also be referred to as a "one-time macro translator." The translator takes a Lotus macro and converts it into a macro written in a different macro language, for use with a different menu tree. Once the translation has been made, a user may run the translated macro as frequently as desired in the other program. Because the macro is written in a different macro language, the program need not refer to a copy of Lotus's menu tree to run the (translated) macro. Thus in *Borland II*, I observed that:

I need not and do not decide whether Borland is prohibited from reading and interpreting macros that have been created by users of 1-2-3. Had Borland created a program that read users' 1-2-3 macros and converted them to macros for use in the Quattro programs' native modes, so that they could be interpreted, executed, modified, debugged, etc. by resort to Borland's command hierarchy, that would have presented a different case from the one now before me.

799 F.Supp. at 214. In other words, I did not decide issues concerning one-time translation of macros into a different macro language, such that the macro could be executed, modified, and debugged without reference to copies of the Lotus menu tree.

Alternatively, consider a continuous macro "interpreter." This facility executes Lotus macros by referring to copies of the Lotus menu tree contained within the program. Macro interpretation may thus be viewed as continuous "translation" of macros as each macro is executed, or "on-the-fly" interpretation. Each time a macro is executed, modified, or debugged, the macro remains written in the Lotus macro language. Each time a macro is executed, modified or debugged, the program refers to the Lotus menu tree or structure.

Both one-time conversion and on-the-fly interpretation may be referred to as "macro translation," as providing an ability to "run" Lotus macros, and as providing "macro compatibility" with Lotus. For both, the user may begin with a Lotus macro and run the macro either by translating it first (one-time translation), or by parsing it with reference to the Lotus menu tree structure each time the user runs the macro (on-the-fly).

An example of the confusion in terms is demonstrated by testimony from Borland's Chief Executive Officer, Phillippe Kahn. Kahn testified that "macro translation was not a viable way to do things." I:13 at 100. From the context of this testimony, I understand the testimony to refer to one-time "macro translation" rather than what Borland now refers to as (on-the-fly) "macro translation" performed by Key Reader.

Borland's Key Reader does not perform one-time translation such as Excel 2.1 does. Rather, Key Reader interprets macros on-the-fly, by reference to "phantom" menus that contain a copy of the Lotus menu tree.

In clarifying this terminology, I do not decide whether copying of the Lotus menu structure for the purpose of one-time translation rather than on-the-fly interpretation should be accorded different treatment under copyright law. Also, I do not decide any issue concerning who (if anyone) owns the copyright in a macro written by a user in reliance on expression found in the Lotus 1-2-3 menu tree, or who (if anyone) owns the copyright in a translation into another language of a macro written in reliance on Lotus 1-2-3. These issues are not before the court. Rather, I make clear that I decide *only* those issues raised by Borland's Key Reader; *ie.*, only issues involved in on-the-fly interpretation using Borland's "phantom" menus.

Having clarified the terminology, I now reject Lotus's argument that on-the-fly macro interpretation does not require copying from the Lotus menu structure and first letters of the command names. To interpret a macro, the program must use the Lotus 1-2-3 menu structure. If a program did not have a representation of the 1-2-3 menu hierarchy somewhere within the program code (or in a file that is used by the code), then there is no way that the program could understand that "rfc" refers to a path through a menu tree to the specific executable operation that changes a cell or cells appearance to monetary units (*i.e.*, a path through the range and format menus to the currency leaf). Whether the menu tree is copied into a file or directly into the code of a program does not make a difference. The scope of copying of the menu tree is the same whether done in one computer language, a different language, or in a file. Accordingly, I reject Lotus's contention that no part of the 1-2-3 menu tree need be copied to interpret 1-2-3 macros.

Nevertheless, I also reject Borland's contention that the menu tree structure and first letters of the menu commands constitute "system" or "method," as those terms are used in copyright law.

Of course, it is possible to think of the Lotus 1-2-3 menu tree as a "system" for translating user keystrokes into executable operations (whether the keystrokes are typed into the keyboard or are taken from the text of a macro command). In the same way, a book may be thought of as a system for communicating various ideas and images to readers. Indeed, the only way of conveying each of the exact images of "Gone With the Wind" in the same order as in "Gone With the Wind" is to copy at least a great deal of the nonliteral expression in the book, if not the precise words of the book as well. Nevertheless, copyright protection extends both to the specific words of "Gone With the Wind" and to nonliteral elements of the novel. Accordingly, the ability to describe a work as a "system" is not decisive of whether the work is a "system," or instead is protected expression, under copyright law.

In calling the copying that is contained in the phantom menu tree a "system," one may attempt to distinguish the copying found in the phantom menus from copying of a book on the following grounds. A book communicates to the reader. The phantom menus, however, permit the user to communicate commands to the program. Thus, one may contend that the phantom menu trees constitute a method or system for the user to communicate executable commands rather than a system (such as a book) for communicating thoughts to the reader.

For the following reasons, I reject this basis for concluding that, in creating the phantom menus, Borland copied a system and not its expression. First, like Lotus's menu tree, the protected expression of a compilation may be viewed as a system for accessing information. The purpose of a compilation is to communicate facts. The specific facts communicated, however, are not copyrightable. Copyright protects only the selection, arrangement and manner of presentation of the facts (to the extent that those elements meet the other requirements for copyrightability). See *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 111 S.Ct. 1282, 113 L.Ed.2d 358 (1991). The selection, arrangement, and manner of presentation in a compilation may provide the user with a *method* or

systematic manner of accessing the (uncopyrightable) facts. Thus, copyright law protects only that part of a compilation that the reader actually *uses* for selection of facts that the reader wants to know. Nevertheless, the expressive aspects of a compilation remain copyrightable.

Accordingly, I conclude that the fact that the copied menu structure and first letters of the menu commands may be used to specify executable operations does not bar a finding that these elements are copyrightable. For the reasons explained in detail in Sections II.A.1 and II.B.2.b of the Phase I Opinion, the structure of the menu tree including its designated keys for invoking commands (*i.e.* what Borland copied into the phantom menus) may also be viewed, in a light favorable to Borland, as a type of selection and arrangement of the executable operations in Lotus 1-2-3. The fact that the phantom menus may be used by a "reader" (whether directly or through the text of a macro) to access Lotus 1-2-3's executable operations does not compel a conclusion that this constitutes a system. See also *Paperback*, 740 F.Supp. at 72-73 (rejecting argument that the macro language is a non-copyrightable element of the program).

Similarly, a program's computer code may be described as a system. Computer code consists of a series of commands that the user issues to the computer by running the program. The commands are written in such a way that the computer code as a whole is a system for communicating the program's functionality to the computer. Nevertheless, Borland cannot and does not dispute that computer code is copyrightable.

Borland's contention that the menu tree structure is a system, or is strictly utilitarian in nature, raises again the same problem of defining the "idea," "system," "process," "procedure," or "method" of Lotus 1-2-3 that I have addressed since the beginning of this case. Borland wishes the court to define the "idea," "system," "process," "procedure," or "method" of Lotus 1-2-3 as including the ability to interpret macros written for use with Lotus 1-2-3. I rejected this contention in deciding the parties' cross motions for summary judgment, see 799 F.Supp. at 212-14, 216-17, and I reject this

argument for the same reasons here. The fact that users of Lotus 1-2-3 have created macros in reliance on expressive aspects of Lotus 1-2-3 does not convert that expression into a part of the "system." That Borland wishes to copy protected expression contained in Lotus's menu tree for what Borland contends is a utilitarian purpose also does not turn that expression into a "system" under copyright law.

See Id at 213-14. *See also Mazer v. Stein*, 347 U.S. 201 [74 S.Ct. 460, 98 L.Ed. 630] (1954) (use of statue as lamp does not bar copyright protection on expression in statue); 1 Nimmer on Copyright § 2.18[D] at 2-207 ("If, however, there is a copying of the copyrightable expression, then an infringement should be found, even if the defendant employs the material for use rather than explanation.").

Borland nevertheless contends that according copyright protection to the menu structure and first letter of the command names would impermissibly protect a "discretionary pattern of events or processes." Section 102(b) provides:

In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work.

17 U.S.C. § 102(b). Borland quotes a proposal by the Information Industry Association to include the following language in Section 102(b):

However, copyright protection may exist in a collection of ideas or abstractions arbitrarily selected from a plurality of alternative ideas or abstractions or in a discretionary pattern of events or processes.

(Docket No. 345 at 8 (quoting Copyright Law Revision: Hearings Before the Subcomm. on Courts, Civil Liberties, and the Admin. of Justice of the House Comm. on the Judiciary, 94th Cong., 1st Sess. 334 (1975))). Because Congress did not

adopt this "exception," Borland contends that Congress manifested an intent that a "discretionary pattern of events or processes" is not protected under copyright law and should be determined to be a "system" under section 102(b).

Nothing Borland presents to the court, however, provides any explanation of why Congress did not include this language in the statute. Failure to adopt this language may manifest a Congressional belief that the language was unnecessary or irrelevant. Indeed, Congress did not adopt any statutory language denying copyright protection for "a collection of ideas or abstractions arbitrarily selected from a plurality of alternative ideas or abstractions or in a discretionary pattern of events or processes."

In any event, the explanatory language accompanying the proposal indicates that the proposal was intended to assure that copyright protection is accorded to nonliteral aspects of computer program code. Thus, Borland's argument may be interpreted as a contention that nonliteral aspects of computer code are not copyrightable, under section 102(b). If so, Borland's argument has been consistently rejected by the courts, including this one.

See Paperback, 740 F.Supp. at 54; Arthur R. Miller, Copyright Protection for Computer Programs, Databases, and Computer Generated Works: Is Anything New Since CONTU?, 106 *Harv.L.Rev.* 978, 994-96 nn. 78-88 (March 1993) (collecting cases). *See also Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 702 (2d Cir.1992) ("We have no reservation in joining the company of those courts that have already ascribed to this logic [that non-literal structures of computer programs are protected by copyright].").

I again reject the argument that copyright cannot extend to nonliteral aspects of a computer program. Moreover, Borland is wrong when it argues that, by failing to adopt the proposed statutory language, Congress manifested any intent concerning the definition of "system" in section 102(b).

Borland next argues that its phantom menus must constitute a "system," or be found "utilitarian" in nature, because the

phantom menus do not appear on the screen and are not communicated to the user.

This argument is readily dismissed. A user may never see computer code, but copyright protection is accorded to the code. Borland proposes a distinction on the ground that computer code may still be printed and read by someone. This distinction is vacuous for three reasons. First, the phantom menus may also be printed out; exhibit 517 is one copy. If one accepts the proposition that nonliteral aspects of computer code are copyrightable, the fact that the printed form of what Borland copied is not identical to any Lotus code or is not actually displayed to the user is not material to a finding that the Lotus menu structure contains copyrightable expression. Second, the fact that the phantom menus are not displayed does not mean that the user does not know they are there. Finally, copyright protection has been accorded to forms of computer code that are not generally intelligible to humans. *See Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1249 (3d Cir. 1983) (object code), *cert. dismissed*, 464 U.S. 1033, 104 S.Ct. 690, 79 L.Ed.2d 158 (1984).

Last, Borland argues that it has removed anything expressive in the Lotus menu tree from its phantom menus, leaving only the first letters of the command names as "markers" of the program's functionality. *See* Docket No. 311 at 25-26. This is not the case. The letters serve as "markers" of the Lotus menu tree structure. Claiming that the "markers" are part of the functionality again incorrectly assumes that the "idea" or "function" of Lotus 1-2-3 includes menu structure and first letters of the command names. Moreover, for the reasons explained in the Phase I Opinion, I conclude that the menu tree structure is original and was not dictated or even substantially limited by functional considerations at the time of its development. Accordingly, I also conclude that the menu structure and organization (including the first letter of the commands, used to mark the structure) are part of the protectable expression found in the Lotus 1-2-3 program.

In sum, I conclude that the Lotus menu structure, organization, and first letters of the command names are not within

the definition of the "idea," "system," "process," "procedure," or "method" of the Lotus 1-2-3 program and constitute part of the protectable expression found in the program.

2. *Borland's arguments concerning earlier "holdings" of this court.*

In arguing that the copying found in Borland's phantom menus is permissible, Borland places considerable reliance on what it characterizes as earlier "holdings of this court that expressly authorize a macro translator such as the Key Reader." *See* Docket No. 345 at 1-3. As a preliminary matter, before the Key Reader complaint in this case, no party before this court had ever pressed any actual dispute or controversy between two parties over whether a specific macro "translation" facility, standing alone, would infringe Lotus's copyrights. Article III courts decide only cases and controversies actually before the court. U.S. Const. Art III, § 2. Accordingly, I could not possibly have "held" in any earlier decision that macro translation as performed by the Key Reader is permissible under copyright law.

After examining Borland's specific citations to my earlier opinions, I also observe that in making this argument Borland has done a bit of interpretative twisting of the language of my earlier opinions. In earlier opinions, I have rejected a defendant's arguments that were based on the incorrect premise that the defendant was somehow required to copy specific elements of the Lotus 1-2-3 program.

See Paperback, 740 F.Supp. at 78 (no showing that copying of entire interface is necessary for macro translation); *Borland II*, 799 F.Supp. at 218 (distinguishing *Crume v. Pacific Mut. Life Ins. Co.*, 140 F.2d 182, 184-85 (7th Cir.), *cert. denied*, 322 U.S. 755[, 64 S.Ct. 1265, 88 L.Ed. 1584] (1944), as involving a system requiring use of only descriptive words, while in this case, varying full-word menu command names is not the only alternative for creating a different expression of the menu tree idea).

Contrary to Borland's arguments, I do not interpret the language in any of my earlier decisions rejecting these arguments as even remotely "authoriz[ing]" Borland to create its Key Reader. Explaining that a defendant's argument is overboard by use of an example is hardly equivalent to "holding" that the example is legally permissible. Moreover, in explaining my earlier decisions, I have twice observed that copying of expressive aspects of Lotus 1-2-3 may not be permissible under copyright law, even if that is the only way to achieve macro compatibility. See *Paperback*, 740 F.Supp. at 69; *Borland II*, 799 F.Supp. at 214. In *Borland II*, I stated that "I need not and do not decide whether Borland is prohibited from reading and interpreting macros that have been created by users of 1-2-3." 799 F.Supp. at 214. In short, I reject Borland's attempt, by selective reading and interpretation of language from earlier opinions, to twist discussions contained in those opinions into "holdings" of this court.

Finally, much of Borland's argument rests on statements concerning whether a "macro translator" such as in *Excel 2.1* infringes. Borland attempts to equate Key Reader with Excel 2.1 by labeling the Key Reader as a "translator." As explained above, however, Borland has expanded the term "translator" from the one-time translation done in Excel 2.1 to include on-the-fly interpretation as done in Borland's programs. In *Borland II*, I expressly did not decide issues concerning one-time translation, 799 F.Supp. at 214 (quoted *supra*), let alone issues concerning an expanded definition of macro "translator"—a definition including on-the-fly interpretation of macros.

D. Substantial Similarity.

For the following reasons, I find that what Borland copied into the Key Reader phantom menus is substantially similar to Lotus 1-2-3.

Contained in the phantom menus is a virtually identical copy of the menu structure and organization of Lotus 1-2-3, using the first letter of command names and other symbols to

delineate that structure. Although Borland's copy may use a different way of representing the menu structure than is contained within the Lotus code or displayed to the user while running Lotus 1-2-3, I find that what was copied by Borland (details of expression of the menu structure) is virtually identical to details of expression of the Lotus 1-2-3 program's menu structure.

I also conclude that differences in the method Borland uses to represent the menu tree structure in its phantom menus file does not negate a finding that the copied expression of the menu tree structure is substantially similar to the Lotus 1-2-3 program. First copyright law protects nonliteral aspects of a copyrighted program. Thus, one need not copy the specific code of a program to infringe copyrights in the program. Second, Borland's copying is analogous to a translation. No identity of words may exist in a translation into French of a book written in English. Nevertheless, the translated copy infringes by making a virtually identical copy of the book, but using a different method for representing the words and grammar structure. See 17 U.S.C. § 101 ("derivative work" includes translation); 17 U.S.C. 106(2) (copyright owner has exclusive right to prepare derivative works). Here, Borland created a virtually identical copy of the Lotus menu structure, but translated (nearly verbatim) the menu structure into a different language for representing menu structures.

Borland next contends that the fact that the phantom menus are not displayed on the screen prevents a finding of substantial similarity. I reject this contention for the same reason that I rejected Borland's argument that Borland copied only unprotected expression because the phantom menus are not displayed. See Section II.C.1, *supra*. That is, copyright law does not require that a program display its source or object code for copied elements of the code (literal or nonliteral) to be substantially similar.

In addition, I find that a user would understand that Borland was using a copy of the Lotus menu structure when executing a macro. Consider, for example, the "/{?}" macro described above. With this macro, the user can traverse the

Lotus menu structure precisely as though typing keys into the Lotus program. Similarly, Key Reader executes macros by sequentially parsing the (Lotus) macro text. This is demonstrated most clearly when one steps through macros in the Borland macro debugger. In addition, Borland's experts contend that a user may not even look to the full command names when using Lotus menus or macros. *E.g.* Liddle Dec'n, VI:11, ¶ 53. I accept this point and reject Borland's premise that the Borland programs do not demonstrate to the user that Borland copied Lotus's menu structure.

Finally, I find that the Lotus menu structure, including the first letters of the command names that mark that structure, constitutes a substantial part of the Lotus expression. What Borland copied into Key Reader is that portion of the Lotus program that determines precisely how the user may select from among the executable operations in the program. Qualitatively, Borland's phantom menus copy a substantial part of what a user would think of as constituting the Lotus program: the menu structure and the sequences of keystrokes for invoking each of Lotus's executable operations.

III. Affirmative Defenses.

Borland raises four affirmative defenses: waiver, laches, estoppel, and fair use. Each is considered below.

A. Waiver.

The parties agree that in order to succeed in its waiver defense, Borland must prove that Lotus voluntarily and intentionally relinquished a known right. *See* Docket No. 311 at 26; Docket No. 338 at 20. That is, Borland must prove (1) voluntary and intentional relinquishment (2) of a right that Lotus was aware that it had.

Borland points to six statements that it contends demonstrate an intention to relinquish any right to assert claims against Key Reader. The first three statements were made before the Key Reader had been introduced into any Borland

product. For these statements, Lotus clearly could not have intended to waive claims specifically against Key Reader; Lotus was unaware of Key Reader. To overcome this obstacle, Borland contends that these statements reflect the intentional relinquishment of a known right to claim that *any* kind of macro "translator" infringes.

The three statements made before Key Reader was released were made by the president of and counsel for Lotus during proceedings in the *Paperback* case. *See* Manzi Dep'n, IX:41; Lemberg Dep'n, IX:24; Trial Transcript, VIII:B.16. Each of the statements concerns whether Lotus intended to assert a claim of infringement against macro translation as performed by Excel.

Borland first cites a statement by Jim Manzi, President of Lotus, made during a deposition in the *Paperback* case. The statement ends with the conclusion that "as far as I know, we have no issue of that [Excel] product." Entirely missing from this statement is any evidence (1) that Manzi was aware of a right to prevent copying found in Excel (even this assumes, without support, that (a) there was copying, and (b) Manzi was aware of that copying), or (2) that Manzi intended to waive a claim against products other than Excel, with respect to any type of macro translation. This statement does not reflect an intention to relinquish any known rights.

Borland next cites statements by Lotus's General Counsel, Thomas Lemberg. The Lemberg statements, in evidence under seal, were not made in a context associated with waiver of rights. Accordingly, I cannot find a manifested intention to relinquish known rights based on these statements. In addition, having examined the Lemberg testimony, I also find that Mr. Lemberg was referring to one-time translation of macros into a different macro language, rather than macro translation in general, and more specifically, rather than referring to any program that executes macros by reference to copies of Lotus's menus. Whether or not the distinction between one-time translation and direct execution of macros makes a material difference under copyright law is not significant; a party may waive rights without following the specific contours of

copyright law. In any event, I find that the Lemberg statements do not demonstrate an intent to relinquish either a specific right or more generally a right to assert claims against all forms of macro translation or execution.

Finally, Borland cites the *Paperback* trial transcript at a point where Lotus's counsel, Henry Gutman, responded to a defendant's contention that the defendant "had to" copy the entire Lotus user interface into its own spreadsheet product. Gutman argued that the premise of this argument is wrong because Excel had made a commercially successful product that did not copy the full interface, and in fact, included the ability to "translate[] and run in Excel" Lotus macros. VIII.B.16 at 45. Based on the materials before me, I find that (1) this statement was intended to discredit a defense argument and does not demonstrate that Lotus intended to waive any right, and (2) this does not demonstrate an intent to waive claims against all types of macro translation as opposed to Lotus's understanding of Excel's one-time translation.

Moreover, each of the three statements cited above was made in the context of an infringement suit against companies that had copied the entirety of Lotus's user interface. Up to the time of the three statements cited by Borland, Lotus had never been faced with issues raised by a program that did not fully copy the Lotus 1-2-3 interface but executed Lotus macros on-the-fly. Accordingly, for each of the statements I find that Borland has not proved (1) that, at the time of the statement, Lotus was aware of a right to bar copying of its menus when the copied menus were used only to execute macros on-the-fly, or (2) that Lotus intended to relinquish its right to assert claims against a program that directly executes Lotus macros by reference to copies of the Lotus menus, rather than by one-time translation of the macros into the product's own macro language.

Borland next cites three statements made by Lotus after Key Reader had been introduced into Borland's products. The first is a response to a Borland request for admissions. According to the response, Lotus admitted that

it does not contend that the . . . ability to execute "macros" originally written using Lotus 1-2-3 by means

of a conversion or translation program, standing alone, would infringe . . . , but qualifies its admission to state that Lotus does contend that the 1-2-3 User Interface of Quattro and Quattro Pro, including, *inter alia*, its use of the 1-2-3 menu structure to provide the ability to execute macros originally written using Lotus 1-2-3, impermissibly copies protected expression contained in Lotus 1-2-3.

See VII:B.1. Later in the same document containing Lotus's response to a request for admission, Lotus objected to a Borland request for an admission that the ability to execute Lotus macros while in Quattro Pro's native mode does not infringe. Lotus's objection was based on the ground that, as far as Lotus knew, Quattro Pro had no such ability. *See* VII:A.7.

This admission does not constitute a waiver of a right to assert infringement claims against Key Reader. Lotus explicitly stated that it *does* contend that use of the Lotus menu structure to provide the ability to execute macros is impermissible, as done in Borland's emulation interface. For the reasons explained in Section II.B above, I find that the Key Reader executes macros by using copies of the Lotus menu structure in a way very similar to the way that the 1-2-3 emulation interface executed macros. The qualification to the admission demonstrates an intent to pursue claims against a program doing precisely what Key Reader does.

Borland contends, however, that the reservation should be interpreted as objecting to a facility such as Key Reader only when it is a part of an emulation interface, and not when it is incorporated into Borland's native menu interface. Considering the circumstances of the admission, however, I find that such an interpretation is unreasonable. First, under Borland's proposed interpretation, the qualification to the admission does not qualify the scope of the admission. The admission without the qualification was already limited to macro translation "standing alone." Second, when specifically asked about running Lotus macros while using the native menu interface, Lotus's answer demonstrated that Lotus was not then aware of Borland's Key Reader feature. It is, therefore,

more reasonable to interpret Lotus's response as (1) a statement that Lotus was not contending in this suit that one-time translation or conversion infringes (a point very distinct from execution of Lotus macros by reference to "phantom" menus), or (2) a statement that Lotus was not contending that translation infringes because there was no need to make that claim in this case—Borland's emulation interface infringed and the scope of that infringement was sufficient for claims against the products then at issue.

If Borland wished later to be in a position to make its present assertion of waiver, it should have made full disclosure up front about Key Reader and obtained an unambiguous response. Borland's present claim of waiver is an attempt, wholly without merit, to escape the consequences of its own litigation strategy. Lotus was entirely free to respond as it did without waiving any right. I find that Lotus's response to the request for admission does not demonstrate an intent to relinquish a known right to assert claims against Key Reader, or a feature like Key Reader.

Borland next points to a statement by Lotus's counsel at a conference in June 1991. *See* VIII:B.17. In making the statement, Lotus's counsel advised the court that if Borland "include[d] 1-2-3 menus and alternate user interface" in a product Lotus believed was about to be released, Lotus would seek a preliminary injunction. Borland's contention that this constitutes a waiver of claims against any program that does not "display" the menu approaches, if not crosses, the line into frivolousness.

Finally, Borland points to a footnote in one of Lotus's briefs. *See* VIII:B.6. Here, Lotus was responding to Borland's contention that it *had* to copy and display the entire menu tree to achieve macro compatibility. In the footnote, Lotus argues that Borland was ignoring *this court's* distinction in *Paperback* between display of menus and executing macros by "interpretation, translation or conversion (as then existed in Excel or the foreign language versions of Lotus 1-2-3)." The foreign language versions of Lotus 1-2-3 perform a one-time translation or conversion of macros from English releases of

1-2-3 into a macro using the foreign language version of 1-2-3. *See* Morgan Dec'n, VI:12. This does not constitute waiver for several independent reasons.

First, the language focuses on the court's distinction, not Lotus's.

Second, Lotus sought to rebut Borland's argument by pointing out that macro compatibility might be achieved without display of the menus and cited examples of one-time macro conversion but not on-the-fly macro interpretation. In the circumstances here, I do not find that impeachment of a defendant's argument by example demonstrates an intent to relinquish any rights.

Third, at the time the argument was made, Borland had not released a product containing Key Reader but without the full emulation interface. Thus, Lotus was not faced with any decision over whether to assert claims against a program that copied Lotus's menus, but only for on-the-fly interpretation. I find that this statement does not demonstrate an intention to relinquish a known right to pursue infringement claims against a program containing Key Reader.

In sum, considering the statements proffered by Borland both separately and as a whole, I find that Borland has not proved that Lotus has waived a right to pursue its claims against Key Reader.

B. Laches.

To succeed in its laches defense, Borland must prove that (1) Lotus inexcusably or unreasonably delayed in raising claims based on Key Reader, and (2) Borland was prejudiced by this delay.

1. Delay.

Lotus first raised claims of infringement based exclusively on Key Reader in its motion for leave to file supplemental complaint, filed December 7, 1992. (Docket No. 250.) Borland contends that this is more than two years after Key Reader was introduced. As explained above, however, Key

Reader was introduced into products that also contained the emulation interface (Quattro Pro 2.0, 3.0, and 4.0).

I begin by examining whether Lotus's delay was unreasonable or inexcusable during the time period when Borland's products included both the emulation interface and Key Reader. I then examine the delay in filing suit after Borland removed the emulation interface and marketed a product containing only Key Reader, *i.e.*, after Quattro Pro 4.01 was announced in August 1992.

For the following reasons, I find that the delay in filing suit before Borland introduced Quattro Pro 4.01 into the market was both reasonable and excusable.

First, even assuming Lotus was aware or should have been aware of Key Reader in Quattro Pro 2.0, 3.0, and 4.0, a fact disputed by Lotus, there is little reason that Lotus should have known (and no evidence that Lotus did know) that Key Reader was anything other than a link to the macro execution facility that is contained in the 1-2-3 emulation interface. In these circumstances, it would have been reasonable for Lotus to believe that if the emulation interface were removed Borland's Key Reader would not function. Indeed, Borland's Chief Executive Officer stated at a deposition in May 1991 that Quattro Pro would not be macro compatible with Lotus 1-2-3 if the 1-2-3 compatible menu tree were removed. *See* I:13 at 52. Thus, claims against the emulation interface might then have appeared to protect Lotus fully, without the added cost of asserting separate claims against Key Reader.

Second, before Quattro Pro 4.01 was released, Lotus was pursuing its strongest claims against all of Borland's existing spreadsheet products, *i.e.*, claims based on the emulation interface. In the circumstances of this case, I cannot say that Lotus was required to assert claims against each feature that Borland added to its (already) infringing products or be faced with the prospect of losing the right ever to assert claims against a program containing that feature. Until the emulation interface was removed, Lotus was not faced with a product that would not be found infringing without a determination that Key Reader infringes. I conclude that it was reasonable

to wait to assert claims against Key Reader until resolution of this dispute was necessary and could be raised explicitly as a live issue.

Third, to the extent Lotus was expressly pursuing claims against macro execution as performed by Borland's emulation interface, *i.e.*, on-the-fly macro interpretation by reference to Lotus's menu structure (*see* VII:B.1, quoted *supra*), Lotus was pursuing claims against something very similar to Key Reader.

Considering these factors, I find that Lotus's failure to bring an action specifically directed at Key Reader for the time period from the incorporation of Key Reader in Quattro Pro 2.0 until release of Quattro Pro 4.01 was both reasonable and excusable because of Lotus's pursuit of infringement claims against the emulation interface in each of Borland's then existing products.

Next, I consider whether Lotus's delay in filing suit after Quattro Pro 4.01 was released was unreasonable or inexcusable. Quattro Pro 4.01 was released after the *Borland II* decision was issued on July 31, 1992. Lotus filed its motion for leave to file supplemental complaint approximately four months after Quattro Pro 4.01 was made public. A few months before Quattro Pro 4.01 was released, Lotus gave notice of an intent to preserve claims against Key Reader (in April 1992, *see* VIII:A.5). Borland acknowledged this notice. (*See* VIII:A.2.) I also note the propriety of taking some time to investigate how Key Reader works before filing a complaint. In light of the brevity of the time period, the nature of investigation of Key Reader that was reasonably required, and Lotus's earlier notice of its intent to preserve claims against Key Reader, I find that Lotus did not unreasonably or inexcusably delay in bringing an action specifically directed at Key Reader.

Further, I reject Borland's contention that the motion to supplement the complaint was filed so late in the proceedings on the original complaint that Lotus's delay was unreasonable. It was the late (and secretive) timing of the release of Quattro Pro 4.01 that resulted in the late timing of Lotus's

claims. Moreover, Lotus agreed to a short time schedule for preparing and presenting its claims for both of the first two Phases of the trial. Accordingly, I find that Lotus's delay in filing claims against Key Reader was reasonable and excusable, despite the advanced stage of proceedings under the original complaint.

2. Prejudice.

As an example of alleged prejudice caused by Lotus's delay in bringing suit, Borland contends that an appeal of copyrightability issues to the First Circuit has been delayed. To make this claim, Borland contends that it removed the emulation interface (but not the Key Reader) in order to expedite an appeal to the First Circuit. The record is barren of evidence to support what counsel contends was Borland's motive for removing the emulation interface.

In addition, even assuming *arguendo* that this unsupported assertion is true, I find that Borland has not proved prejudice. For the reasons explained on the record at the September 23, 1992 conference, I denied Borland's motion for interlocutory appeal. This was before Lotus moved for leave to include claims against Key Reader in this case. Thus, before Lotus filed its motion for leave to supplement, I had already concluded that an interlocutory appeal was not appropriate. Borland has not demonstrated that it could have secured an interlocutory appeal of the summary judgment decision, even if Lotus never asserted claims against Key Reader.

To demonstrate prejudice, Borland also points to costs in developing and advertising Key Reader. Even assuming these costs were appreciable, a finding that Borland provides little evidence to support, I find that these costs were not incurred as a result of any delay by Lotus. Of course, Lotus had not delayed at all in bringing suit before Key Reader was introduced into a Borland product. In any event, as to Quattro Pro 2.0, 3.0, and 4.0, these products have been found to infringe whether or not Key Reader was present. Borland points to no costs with respect to versions 2.0, 3.0, and 4.0 that might have been avoided had Lotus asserted infringement claims directed

at Key Reader earlier. Moreover, Borland did not remove the emulation interface from Quattro Pro 2.0, 3.0, or 4.0, all released after Lotus had filed the complaint in this action. I find that Borland would have included Key Reader along with the emulation interface in its products whether or not Lotus immediately asserted that Key Reader infringes.

With respect to Quattro Pro 4.01 and later releases, Borland released these programs months after Lotus had sent a message (that Borland had acknowledged) that Lotus was seeking to preserve claims against programs containing Key Reader. (See discussion in preceding section.) I conclude that any prejudice suffered by Borland's inclusion of Key Reader in Quattro Pro 4.01 and later releases was not caused by delay on Lotus's part.

C. Estoppel.

As was true with respect to the estoppel claims in Phase I, the parties apparently agree that to establish a defense of estoppel in relation to Phase II issues, Borland must prove that Lotus engaged in (1) conduct that induced Borland to change its position in good faith, or (2) conduct on which a reasonable person would rely. As I did in the Phase I Opinion, again without endorsing this specific formulation of the standard for deciding a claim of estoppel, I proceed, to Borland's advantage, to apply the standard stated by the parties.

In determining whether Borland has proved reasonable reliance on any conduct by Lotus, I consider first Borland's inclusion of Key Reader in Quattro Pro versions 2.0, 3.0, and 4.0, then turn to inclusion of Key Reader in Quattro Pro 4.01 and later releases.

Versions 2.0, 3.0, and 4.0 were released after Lotus filed suit claiming that Quattro and Quattro Pro infringe. Nevertheless, Borland included the emulation interface in each of these products. Accordingly I find that, even if Lotus had clearly and unequivocally stated an intention to assert claims against something like Key Reader, Borland would have

included Key Reader in these products anyway. Borland did not rely on any statement by Lotus in developing Key Reader.

For the following reasons, I also find that Borland has not proved that a reasonable person would have relied on any conduct by Lotus in developing or advertising Key Reader in versions 2.0, 3.0, and 4.0.

Borland contends first that it relied on Lotus's original complaint, ¶¶ 17-19, to conclude that Lotus would never assert claims against a program that copied Lotus's menu structure within the computer code but did not display the Lotus menu structure on the screen. This contention is without merit. The complaint alleged claims against products that did not have a Key Reader. The complaint focused on the emulation "user interface" because that is precisely the portion of the programs (in existence at the time the complaint was drafted), both as displayed on the screen and as implemented by the computer code, that infringes. In any event, without Borland's computer code, there is no Borland emulation interface. The complaint cannot be read in any way other than as asserting claims for infringement based on Borland's computer code. Finally, the premise to Borland's argument, that Lotus knew or should have known enough about the computer coding of Quattro and Quattro Pro to make specific allegations in the complaint about the computer code is without support in the record. Borland's claim that it reasonably relied on the original complaint as demonstrating an intention by Lotus to pursue infringement claims based only on Borland's display of the interface and not on Borland's code is baseless.

Borland next cites three statements, made after Key Reader was first released in Quattro Pro 2.0, by Lotus executives or counsel. These three statements are the same statements that Borland cited in its waiver argument, *i.e.*, (1) the admission, (2) the statement at the June 1991 conference, and (3) a footnote in one of Lotus's briefs. For the reasons explained in Section III.A, *supra* I conclude that Borland could not reasonably rely on any of these statements as demonstrating an intent not to pursue claims against Key Reader. Moreover, to

the extent that the statements and the definition of macro "translation" are ambiguous, reliance on those statements would be unreasonable. A reasonable person or company in Borland's position would have sought clarification before relying on the statements as a position adopted by Lotus.

In addition, each of these statements was made after Key Reader was introduced into Quattro Pro 2.0. Borland, when originally developing and introducing Key Reader, could not have relied on statements made later, after Key Reader had been developed. Nevertheless, Borland contends that it relied on these statements in retaining Key Reader in Quattro Pro 4.01 and later releases. As explained above, however, in April 1992, Lotus clearly stated (and Borland acknowledged) that Lotus wished to preserve claims against Key Reader. In light of this, I find that for Quattro Pro 4.01 and later releases, Borland could not have reasonably relied on the cited statements as demonstrating an intent by Lotus not to assert claims against Key Reader.

Finally, Borland's General Counsel avers that, when recommending to keep Key Reader in the program, he relied "most directly" on the court's language in the *Borland II* summary judgement opinion. *See* VI:9, ¶¶ 13-14. First, for the reasons explained in Section II.C.2, I find that Borland's selective reading of the opinion and twisted interpretation of the court's language is unreasonable; reliance on this language, therefore, also would be unreasonable. Second, I am not aware of any precedent for basing an estoppel defense not on conduct of the party alleged to be estopped, but on conduct of the court. Borland offers no authority to support this proposition. Instead, Borland contends that the court's language reflects Lotus's arguments. I reject this contention. To support such a contention, Borland would have to place in evidence Lotus's arguments and conduct, rather than simply pointing to the court's decision. I have examined the language Borland points to in Lotus's earlier briefs and argument and reject Borland's contention that Borland reasonably relied on Lotus's statements or conduct.

In any event, to the extent Borland relied on the court's language in deciding to include Key Reader in Quattro Pro 4.01, Borland was not relying on Lotus's conduct.

Having considered the several instances of conduct cited by Borland both individually and as a whole, including any delay in filing suit, I find that Borland has not proved its estoppel defense.

D. Fair Use.

Borland contends that copying of the Lotus menu structure and first letters of the menu commands is a fair use of Lotus's copyrighted programs. The fair use doctrine is an "equitable rule of reason." *Sony Corp. of Am. v. Universal City Studios, Inc.*, 464 U.S. 417, 448, 104 S.Ct. 774, 792, 78 L.Ed.2d 574 (1984) (quoting legislative history). The doctrine

permits courts to avoid rigid application of the copyright statute when, on occasion, it would stifle the very creativity which that law is designed to foster.

Stewart v. Abend, 495 U.S. 207, 236, 110 S.Ct. 1750, 1768, 109 L.Ed.2d 184 (1990) (quoting *Iowa State University Research Found. Inc. v. American Broadcasting Cos.*, 621 F.2d 57, 60 (2d Cir.1980)).

Congress set forth four statutory factors to guide consideration of what constitutes a fair use. Section 107 of the Copyright Act provides that:

the fair use of a copyrighted work . . . for purposes such as criticism, comment, news reporting, teaching . . . , scholarship, or research, is not an infringement of copyright. In determining whether the use made of a work in any particular case is a fair use the factors to be considered shall include—

- (1) the purpose and character of the use, including whether such use is of a commercial nature or is for non-profit educational purposes;
- (2) the nature of the copyrighted work;

(3) the amount and substantiality of the portion used in relation to the copyrighted work as a whole; and

(4) the effect of the use upon the potential market for or value of the copyrighted work.

17 U.S.C. § 107. These factors are not exclusive. Each case is to be decided on its own facts. *See Harper & Row, Publishers, Inc. v. Nation Enters.*, 471 U.S. 539, 560, 105 S.Ct. 2218, 2230, 85 L.Ed.2d 588 (1985).

1. "[P]urpose and character of the use."

The first statutory factor focuses primarily on whether the copied materials are used to secure profit.

The crux of the profit/nonprofit distinction is not whether the sole motive of the use is monetary gain but whether the user stands to profit from exploitation of the copyrighted material without paying the customary price.

Harper & Row, 471 U.S. at 562, 105 S.Ct. at 2231. Section 107(1) contrasts a use that is "of a commercial nature" with a use that is for "nonprofit educational purposes." According to the preamble in section 107, copying for "criticism, comment, news reporting, teaching . . . , scholarship, or research may be closer to being a "nonprofit educational use" than a commercial use.

In the instant case, there can be little dispute that Borland copied from Lotus for "commercial" purposes. Borland sells its spreadsheet products for profit. The Key Reader is included to make the program more attractive to potential customers than have Lotus 1-2-3 macros. Borland has copied Lotus's menu structure in order to enhance profits from sales of its spreadsheet products, without having secured a license to do so.

Borland tries to minimize the impact of having a profit motive on its fair use defense by contending that Key Reader is actually used by consumers and only for interpreting the macros that the consumers have written. (Docket 345 at 19.) The consumer's use of the part of Quattro Pro that Borland

copied from Lotus, however, does not change the character of Borland's use of the copied materials—to please consumers and increase sales. To contend that copying is closer to a fair use because customers want the copied materials is entirely without merit. “It is fundamentally at odds with the scheme of copyright to accord lesser rights in those works that are of the greatest importance to the public.” *Harper & Row*, 471 U.S. at 559, 105 S.Ct. at 2230.

Borland nevertheless contends that the present case is “almost identical” to *Lewis Galoob Toys, Inc. v. Nintendo of Am., Inc.*, 964 F.2d 965 (9th Cir.1992), cert. denied, ___ U.S. ___, 113 S.Ct. 1582, 123 L.Ed.2d 149 (1993). In *Galoob*, the alleged infringer sold a product (the “Game Genie”) that allowed private users of Nintendo's copyrighted games to use Nintendo's games in an enhanced fashion. Nintendo did not argue that the Game Genie directly infringes Nintendo's copyright. Rather, Nintendo argued that use of the Game Genie with a Nintendo game creates a derivative work of the copyrighted Nintendo game. See *id.* at 970. Thus, Nintendo's infringement claims concerned only whether Galoob contributed to a consumer's direct infringement of Nintendo's copyright when the consumer used the Game Genie with a Nintendo game at home. For this reason, the opinion concerning fair use explicitly restricted its focus to whether a consumer's private use of the Game Genie was a fair use (and not on Galoob's use of the Game Genie—sale to the consumer for profit). *Id.* Private use by a consumer is a nonprofit use rather than a commercial use. See *id.* See also *Sony Corp.*, 464 U.S. at 449, 104 S.Ct. at 792.

The present case is readily distinguished from *Galoob*. Lotus contends and I have found that Borland copied and that the Quattro Pro products, including Key Reader, directly infringe Lotus's copyright. The fair use issues in this case properly focus on Borland's use (and not the consumer's use) of the infringing products. Otherwise, a book store could copy and sell “Gone With the Wind” without permission. The book store would rely on the consumer's private use of the copied material to claim that its copying is a fair use.

Borland cites *Sega Enters. Ltd v. Accolade, Inc.*, 977 F.2d 1510, 1522 (9th Cir.1992) for the proposition that “the commercial nature of a use is a matter of degree, not an absolute.” In finding that the commercial nature of the defendant's use in *Sega* was of a lesser degree, the Ninth Circuit relied on the fact that the copied materials were not included in defendant's final product. *Id.* Here, however, what Borland copied from Lotus is included in the product it sells to the public.

Borland also relies on *Sega* to argue that Borland has performed its own creative work in developing Quattro Pro, and therefore, Key Reader merely “supplements” Borland's own creative work. Assuming this to be true does not change the character of Borland's commercial use of the copied material. Most of Borland's own creative work, including its own menu structure, exists in Borland's programs independently of whether or not Borland includes a Key Reader in its product. Borland used the copied materials to please customers and increase sales, independently of the creative work it did. In addition, the *Sega* case is readily distinguished. *Sega* involved “intermediate” copying; the defendant did not include copied materials in the final product. 977 F.2d at 1522. Here, Borland “supplemented” the product it sells to consumers with a facility including the impermissibly copied materials.

In sum, I find that the “purpose and character” of Borland's use of Key Reader is entirely commercial. This conclusion concerns just a single factor among a number of factors a court weighs in deciding a fair use defense. Nevertheless, when a defendant's use of copied materials is determined to be commercial, that use is “presumptively an unfair exploitation of the monopoly privilege that belongs to the owner of the copyright.” *Abend*, 495 U.S. at 237, 110 S.Ct. at 1768 (quoting *Sony Corp.*, 464 U.S. at 451, 104 S.Ct. at 793). See also *Harper & Row*, 471 U.S. at 562, 105 S.Ct. at 2231.

2. “[T]he nature of the copyrighted work.”

This statutory factor focuses on locating a work, according to its nature, on the spectrum from factual works to fiction or fantasy. “The law generally recognizes a greater need to

disseminate factual works than works of fiction or fantasy.” *Harper & Row*, 471 U.S. at 563, 105 S.Ct. at 2232. Similarly, a work largely dictated by functional considerations is closer to the “factual works” end of the spectrum than the fiction end. *See Sega*, 977 F.2d at 1524.

Also, even for factual works,

there are gradations as to the relative proportion of fact and fantasy. One may move from sparsely embellished maps and directories to elegantly written biography. The extent to which one must permit expressive language to be copied, in order to assure dissemination of the underlying facts, will thus vary from case to case.

Harper & Row, 471 U.S. at 563, 105 S.Ct. at 2232 (quoting Gorman, *Fact or Fancy? The Implications for Copyright*, 29 J. Copyright Soc. 560, 561 (1982)). Thus, weighing this factor requires a determination of how much freedom of expression the author had in formulating the copyrighted expression. If there was little freedom, the expression is like a bare map. With greater expression found in the work, the scope of copyright protection is greater and a finding of fair use less likely.

For the reasons explained in the Phase I Opinion, I concluded that Lotus had substantial freedom of expression in forming its menu tree. The degree of freedom of expression in creating a menu structure using only first letters of command names is no less than the degree of freedom in designing Lotus’s menu tree using full command names. Nevertheless, the menu tree is based on the set of executable operations selected for Lotus 1-2-3. Accordingly, I find that the copyrighted work in this case is similar to a “factual work,” but that copying from the substantial expression found in the menu tree is not remotely necessary for disseminating the underlying executable operations (or “facts”). I find that this factor does not weigh significantly in either party’s favor.

Borland contends that the “utilitarian nature” of the menu tree weighs in favor of a finding of fair use. Lotus’s copyright does not, however, extend to aspects of the program that are utilitarian or functional and not expressive. Moreover, Bor-

land’s reliance on *Sega* is misplaced. The *Sega* court determined that copying of protected expression was necessary in order to gain access to unprotected aspects of the program. For this reason, the *Sega* court determined that copyright protection for the code in *Sega* was entitled to a lower degree of protection than other literary works (*i.e.*, less protection against the “intermediate” copying done by the defendant in that case). 977 F.2d at 1526. Here, Borland’s copying was not necessary to examine unprotected aspects of Lotus 1-2-3.

3. “[A]mount and substantiality of the portion used in relation to the copyrighted work as a whole.”

Borland created a virtually identical copy of the menu structure and first letters of command names from Lotus 1-2-3’s menu tree. This necessarily includes copying of the keystroke sequences used by Lotus to access executable commands and Lotus’s macro language. For the reasons explained in Section II.D, *supra*, I find that this is a substantial part of the copyrighted expression in Lotus 1-2-3.

Borland argues that “it is difficult to see how Borland could use even less of 1-2-3.” Docket No. 345 at 21-22. The answer to this is simple. Borland could have copied none of Lotus’s expression, or (though I do not comment on the permissibility of doing so) less than the complete 1-2-3 menu structure and first letters of command names.

The fact that Borland uses the menu tree structure and first letters of the command names only for macro translation is not significant in evaluating this statutory factor. “[A] taking may not be excused merely because it is insubstantial with respect to the *infringing* work.” *Harper & Row*, 471 U.S. at 565, 105 S.Ct. at 2233 (original emphasis).

I find that this statutory factor weighs in favor of Lotus.

4. “[T]he effect of the use upon the potential market for or value of the copyrighted work.”

The fourth statutory factor focuses on whether “some meaningful likelihood of future harm exists.” *Sony Corp.*, 464 U.S. at 451, 104 S.Ct. at 793 (original emphasis). To negate a claim

of fair use, a copyright holder may show either "that the particular use is harmful, or that if it should become widespread, it would adversely affect the potential market for the copyrighted work." *Id.* (holding that, for a noncommercial use, copyright holder must make this showing). "This inquiry must take account not only of harm to the original but also of harm to the market for derivative works." *Harper & Row*, 471 U.S. at 568, 105 S.Ct. at 2234.

As noted in Section III.D.1, *supra*, a commercial use of the copied material is presumptively unfair. Thus, "[i]f the intended use is for commercial gain, [the] likelihood [of future harm] may be presumed." *Sony Corp.*, 464 U.S. at 451, 104 S.Ct. at 793.

In the present case, I find that there is a meaningful likelihood that Borland's copying negatively affects the market for and value of Lotus's copyright. Quattro Pro's functionality is largely equivalent to Lotus 1-2-3's functionality. The operations the programs perform on spreadsheets are similar in nature. From this, I infer that many users will not go to the expense of purchasing, maintaining, and updating both the Quattro Pro and Lotus lines of spreadsheet products.

Borland's inclusion of Key Reader caters to users who already own Lotus 1-2-3; *i.e.*, users that have developed macros for use with Lotus 1-2-3. This may directly affect sales of Lotus 1-2-3 programs. For example, a company using Lotus 1-2-3 may decide to shift to use of Quattro Pro rather than purchasing additional copies of Lotus 1-2-3 for new employees. Inclusion of Key Reader may also have an impact on sales of updates to or new releases of Lotus 1-2-3. For example, Key Reader increases the likelihood that a user will shift to Quattro Pro rather than purchase an update to Lotus 1-2-3.

For these reasons, I infer that there is a meaningful likelihood of harm to Lotus (by a negative impact on the market and a decrease in value of Lotus's copyright) stemming from Borland's use of Key Reader.

Borland seeks to rebut this inference with evidence that Lotus's share of the spreadsheet market has remained steady

from the time before Borland introduced Key Reader to the present. I reject this argument for the following reasons.

First, this statutory factor includes an examination of what would happen "if [the use] should become widespread." *Sony Corp.*, 464 U.S. at 451, 104 S.Ct. at 793. This is a necessary consideration. Otherwise, a local bookstore would be free under the fair use doctrine to copy and sell a popular book (without license) because the impact on the national market is negligible. Here, if consumers widely purchase Quattro Pro intending to use Key Reader, I find that the market for Lotus 1-2-3 products and value of the Lotus copyright would be diminished.

Second, Borland provides no evidence that maintaining market share is equivalent to having a copyright of undiminished value. A number of factors other than a complete lack of impact on the market for Lotus 1-2-3 may explain a constant market share. For example, pricing of both 1-2-3 and competing products (including cut prices or lack of increase in prices of 1-2-3) may maintain a constant market share. Also, advertising might be increased to maintain market share. The evidence presented by Borland falls far short of supporting an inference that Lotus market has not been adversely affected.

Third, even if Lotus can maintain a constant market share without increased costs associated with lower prices or higher expenses, this does not demonstrate that the copying found in Key Reader has not diminished Lotus's market share from what it would be if Borland had not copied. Without Borland's copying, Lotus's market share may have increased.

In short, I find that Borland's limited evidence concerning market share is entirely insufficient to demonstrate that Lotus has not been harmed or to rebut the presumption that Borland's commercial use of Key Reader presents a meaningful likelihood that Lotus will be harmed. This statutory factor weighs in favor of Lotus.

5. Other factors.

The statutory list of factors a court may consider in deciding the merits of a fair use defense is not exclusive. As an additional factor, Borland contends that public policy dictates that Borland's use of Key Reader be considered a fair use. Borland argues that permitting Borland to market an "improved" spreadsheet product while maintaining macro compatibility for users of Lotus 1-2-3 stimulates (Borland's) artistic creativity. In essence, Borland contends that, when forming new artistic expressions, Borland should be permitted to tap into the market for existing expression by copying not just the idea but by copying from the expression of the idea. I reject Borland's arguments for the following reasons.

First, this "new" factor is just another argument concerning the "purpose and character" of Borland's use of the copied expression. Thus, I reject Borland's argument for the same reasons that I concluded that Borland's use of the copied materials is commercial. Borland is not entitled to rely on the consumers' private use of the program for execution of macros to claim fair use.

Second, even when barred from copying expression, Borland was and is free to create new expressions of the ideas contained in Lotus 1-2-3. Borland presents no evidence that providing users with macro compatibility, or the ability to execute macros by reference to copies of Lotus's menus, is necessary to permit copying of the ideas found in Lotus 1-2-3. Compare with *Sega*, discussed *supra*. Borland has not demonstrated that allowing it to copy Lotus's expression would stimulate creativity more than the general scheme of providing copyright protection for an author's expression.

6. Summary.

Each of the factors to be weighed in deciding whether Borland's copying constitutes a fair use either weighs against a finding of fair use, or is entitled to little or no weight. Considering all factors, I find that Borland's copying is not a fair use of Lotus's copyrighted expression.

IV. Summary.

For the reasons explained above, I find that Borland's Key Reader infringes Lotus's copyrights. Further, I find that (1) Lotus has not waived claims against Key Reader, (2) Lotus is not barred by laches or estoppel from asserting claims against Key Reader, and (3) including expression copied from Lotus 1-2-3 in Borland's Key Reader facility is not a fair use of the protected expression.

Permanent Injunction

For the reasons stated in the July 31, 1992 Memorandum and Order, and for the reasons stated and on the Findings and Conclusions recited in the Opinion of June 30, 1993, as amended August 19, 1993 (the "Phase I Opinion"), and the Opinion of August 12, 1993, as amended August 19, 1993 (the "Phase II Opinion"), it is ORDERED, ADJUDGED, AND DECREED:

1. Defendant Borland International, Incorporated ("Borland") has infringed plaintiff's copyrights in the computer software program "Lotus 1-2-3" by its development, manufacture and sale of: (1) Quattro and Quattro Pro version 1.0, by Borland's inclusion of its "1-2-3 emulation" interface; (2) Quattro Pro versions 2.0, 3.0, and 4.0, by Borland's inclusion of its "1-2-3 emulation" interface and its "Key Reader" facility; and (3) Quattro Pro SE, 4.01 and Quattro Pro for Windows by Borland's inclusion of its "Key Reader" facility.

2. Acting under 17 U.S.C. § 502(a), the court orders that Borland, its officers, agents, servants, employees and attorneys, and all persons in active concert or participation with them, are hereby permanently enjoined from manufacturing, selling, distributing, licensing, or continuing to license for manufacture, sale, distribution or sub-licensing

(i) Quattro, Quattro Pro versions 1.0, 2.0, 3.0, 4.0, SE, 4.01, and all versions of Quattro Pro for Windows that

have been or are on the market on this date (August 19, 1993) ("Quattro Pro for Windows"), and

(ii) any work that contains in any portion, component or module thereof, a copy of the Lotus 1-2-3 menu commands and/or menu structure, in any form. Borland may at any time, however, apply to this court for modification of this clause upon a showing of good cause for determining that Borland has developed or proposes to develop a product that is within this clause but for special reasons

(a) the product is not infringing, or

(b) an alternative form of remedy allowing marketing on specified conditions, including payment of royalties, is more appropriate than an injunction.

3. In addition, upon entry of this Permanent Injunction, Borland shall forthwith take all necessary steps to terminate any existing licenses, distribution agreements or other arrangements pursuant to which it has manufactured, sold, distributed or licensed any of Quattro, Quattro Pro version 1.0, 2.0, 3.0, 4.0, SE, 4.01 or Quattro Pro for Windows and it shall, at its own expense, on or before August 27, 1993, notify each and every other party to any such license, distribution agreement or other arrangement of the existence and terms of this Permanent Injunction. Borland shall take all reasonable measures necessary to ensure, so far as it can control, that all such manufacturers, distributors and resellers comply with the terms hereof, including reasonable measures to prevent the selling of infringing products, which may include, but does not require, repurchase of products, and which does not include a general recall.

UNITED STATES DISTRICT COURT
D. MASSACHUSETTS

Civ. A. No. 90-11662-K

June 30, 1993

As Amended Aug. 19, 1993

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—v.—

BORLAND INTERNATIONAL, INC.,

Defendant.

James C. Burling, Jeffrey B. Rudman, Hale & Dorr, Boston, MA, Henry B. Gutman, Kerry L. Konrad, O'Sullivan, Graev & Karabell, New York City, *for plaintiff.*

Laura Steinberg, Sullivan & Worcester, Boston, MA, Lynn H. Pasahow, McCutchen, Doyle, Brown & Enersen, San Francisco, CA, David L. Hayes, Mitchell Zimmerman, Fenwick & West, Palo Alto, CA, Peter E. Gelhaar, Donnelly, Conroy & Gelhaar, Boston, MA, Gary L. Reback, Nina F. Locker, Peter N. Detkin, David A. Priebe, James A. DiBoise, Wilson, Sonsini, Goodrich & Rosati, Palo Alto, CA, Laurence H. Tribe, Cambridge, MA, *for defendant.*

OPINION

KEETON, *District Judge.*

On July 31, 1992, the court allowed, in part, a motion for summary judgment filed by plaintiff Lotus Development Corporation ("Lotus") and denied the cross motion for summary judgment by defendant Borland International, Incorporated ("Borland"). After extended procedural maneuvering, the parties agreed to try remaining liability issues without a jury. An explanation of the proceedings leading up to the trial is essential to precise identification of the issues raised by the parties in the nonjury trial of February 1-3 and March 31-April 2, 1993.

I. Earlier Proceedings.

A. Partial Summary Judgment.

This Opinion assumes the reader's familiarity with the Memorandum and Order allowing, in part, Lotus's motion for summary judgment. That document was published as *Lotus Dev. Corp. v. Borland Int'l Inc.*, 799 F.Supp. 203 (D.Mass.1992) (July 31 Memorandum and Order). In addition, the terminology used in this Opinion follows the terminology set forth in detail in the earlier Memorandum and Order. *Id.* at 206-208. Background information appears in two earlier documents issued by this court. The first is a Memorandum and Order in this case, *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 788 F.Supp. 78 (D.Mass.1992). The second is an opinion in a related case involving claims of infringement of copyrights for the Lotus 1-2-3 program. *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F.Supp. 37 (D.Mass.1990).

The July 31 Memorandum and Order explained the standard to be applied in this case for determining copyrightability issues:

FIRST, in making the determination of "copyrightability," the decisionmaker must focus upon alternatives

that counsel may suggest, or the court may conceive, along the scale from the most generalized conception to the most particularized, and choose some formulation, some conception of the "idea," "system," "process," "procedure," or "method"—for the purpose of distinguishing between the idea, system, process, procedure, or method and its expression.

* * *

SECOND, the decisionmaker must focus upon whether an alleged expression of the idea, system, process, procedure, or method is limited to elements essential to expression of that idea, system, process, procedure, or method (or is one of only a few ways of expressing the idea, system, process, procedure, or method) or instead includes identifiable elements of expression not essential to every expression of that idea, system, process, procedure, or method.

THIRD, having identified elements of expression not essential to every expression of the idea, system, process, procedure, or method, the decisionmaker must focus on whether those expressive elements, taken together, are a substantial part of the allegedly copyrightable "work."

Borland, 799 F.Supp. at 211 (quoting *Borland*, 788 F.Supp. at 90 (quoting *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F.Supp. 37, 60-61 (D.Mass.1990)) (all emphases omitted)).

Applying this test, I determined that the "idea," "system," "process," "procedure," or "method" of the Lotus 1-2-3 program is a menu-driven electronic spreadsheet whose

user interface involves a system of menus, each menu consisting of less than a dozen commands, arranged hierarchically, forming a tree in which the main menu is the root/trunk of the tree and submenus branch off from higher menus, each submenu being linked to a higher menu by operation of a command, so that all the specific spreadsheet operations available in Lotus 1-2-3 are

accessible through the paths of the menu command hierarchy.

Id at 216-17. (The meanings of menu command, menu structure and menu tree are explained at greater length below.) I concluded also that

as a matter of law, Borland's Quattro products infringe the Lotus 1-2-3 copyright because of (1) the extent of copying of the "menu commands" and "menu structure" that is not *genuinely* disputed in this case, (2) the extent to which the copied elements of the "menu commands" and "menu structure" contain expressive aspects separable from the functions of the "menu commands" and "menu structure," and (3) the scope of those copied expressive aspects as an integral part of Lotus 1-2-3.

Id at 223 (original emphasis).

Nevertheless, I concluded that determining the scope of relief in this case depends on resolution of disputed factual contentions because Lotus contended and Borland disputed "that the copying of separable expressive elements of the Lotus 1-2-3 user interface into the Quattro programs was greater than the minimum essential to constituting a substantial part of the Lotus 1-2-3 work," which I had determined on motion for summary judgment not to be genuinely in dispute. *Id*. In other words, there is no genuine dispute of fact that the Quattro and Quattro Pro programs infringe, but fact issues remain as to the scope of impermissible copying. Specific fact issues apparent on the record at that time concerned (1) whether Borland copied the long prompts of Lotus 1-2-3, (2) whether the long prompts contain expressive elements, and (3) the extent (if any) that functional constraints limit the number of possible ways that the Lotus menu command hierarchy could have been arranged at the time of its creation. *See* Order Regulating Jury Trial, September 30, 1992 (Docket No. 232) at 20.

In addition, I concluded that Lotus was entitled to summary judgment against Borland on the affirmative defense of

waiver, but not on the affirmative defenses of laches and estoppel. *See Borland*, 799 F.Supp. at 222-23.

B. Further Proceedings Before Trial.

Up to the time of the court's ruling of July 31, 1992, the parties' contentions concerned issues raised in the allegations of the "original complaint" filed July 2, 1990, concerning infringement of Lotus 1-2-3 by Quattro and Quattro Pro's "emulation interface." Trial for the remaining liability issues in the original complaint was scheduled before a jury. In January 1993, this court permitted Lotus to file a supplemental complaint alleging copyright infringement by Borland in programs containing a "Key Reader" feature (which is described in some detail in the August 12, 1993 Opinion that addresses the issues of the "Key Reader" phase of the trial)—specifically in Quattro Pro versions 2.0, 3.0, 4.0, SE, 4.01 and Quattro Pro for Windows. *See* Docket 250, Exh. A.

After these developments and as the parties were preparing for trial of issues raised in the original complaint, the parties entered into a series of stipulations that altered the nature of the proceedings. *See* Stipulation and Order Regulating Trial (Docket No. 330); Stipulation and Order Regulating Key Reader Trial (Docket No. 349).

The first set of stipulations concerned trial of issues raised in the original complaint. These stipulations

govern the trial of all issues not previously finally decided by way of summary judgment concerning Borland's alleged liability herein, and all its defenses thereto, *excluding* the issues raised by Lotus' Supplemental Complaint concerning the "Key Reader" feature (the "Phase I Trial").

Docket No. 330, ¶ 1. With respect to issues raised in the original complaint, the parties waived jury trial for the liability issues that had previously been scheduled for the Phase I trial.

With respect to the long prompts, the parties stipulated that:

1. The order of display of the long prompts within the "1-2-3-compatible" modes of Quattro and Quattro Pro

follows the order of display of the menu commands within those modes, and each such long prompt provides a short textual description of the command to which it relates.

2. Lotus shall not contend, in this action or any appeal therefrom, that Borland has copied the long prompts of Lotus 1-2-3 in Quattro or Quattro Pro.
3. Borland shall not contend, in this action or any appeal therefrom, that Borland has not copied the long prompts of Lotus 1-2-3 in Quattro or Quattro Pro.
4. Neither party shall contend, in this action or any appeal therefrom, that the issue of whether or not Borland copied the long prompts of Lotus 1-2-3 in either Quattro or Quattro Pro is material to any other issue that has been or will be resolved in this case.

Docket No. 330, Exh. A, ¶¶ 1-4.

The second set of stipulations "govern[s] the trial of all liability issues (including any defenses thereto) raised by Lotus' Supplemental Complaint concerning the 'Key Reader' feature (the 'Phase II Trial')." Docket No. 349, ¶ 1. For trial of the liability issues raised by the supplemental complaint, the parties waived their rights to trial by jury. *Id.*

C. Summary of Issues Before the Court.

Phase I of the trial was held on February 1-3, 1993. At that time, the issues before the court were the scope of infringement by Borland and Borland's affirmative defenses of laches and estoppel (the affirmative defense of waiver having been resolved at summary judgment). After the close of Borland's evidence, however, Borland was allowed leave to amend its answer to assert an affirmative defense of fair use. *See* Memorandum and Order, March 30, 1993 (Docket No. 353). In response, Lotus moved for judgment on partial findings. *See* Fed.R.Civ.P. 52(c). After hearing in open court and for the reasons stated on the record, I allowed Lotus's motion for

judgment on the issue of Borland's fair use defense to the original complaint (Phase I).

Phase II of the liability trial was held on March 31-April 2, 1993. The issues presented to the court included the full range of liability determinations for the "Key Reader" supplemental complaint. In addition, the parties tried Borland's defenses (to the "Key Reader" complaint) of waiver, laches, estoppel, and fair use.

This opinion addresses the issues raised in Phase I of the trial. On August 12, 1993, the court released an Opinion resolving the issues raised in Phase II.

II. Scope of Infringement in Phase I Trial.

As I understand the parties' stipulations and arguments, Lotus does not now contend that Borland copied the entire 1-2-3 interface. Rather, Lotus claims that Borland has illegally copied the Lotus 1-2-3 "menu commands" and "menu structure." Accordingly, the only issues before the court concern copying of the menu commands and structure.

Borland contends that copying of menu commands and structure is permissible because of functional constraints on formulation of the menu commands and structure. To the extent that Borland contends that the menu commands and structure as a whole are not copyrightable, Borland's contention was rejected as a matter of law at summary judgment. Nothing Borland presented at the Phase I trial alters my view that there is no genuine dispute that "a large part of the structure and arrangement of the menu commands is not driven entirely by functional considerations." 799 F.Supp. at 218.

Although I determined that no genuine dispute of fact had to be resolved in order to determine that the menu commands and structure contain protectable expression, I also determined that disputed factual contentions might have to be resolved to determine the *scope* of infringement. Among factors bearing on the scope of infringement are (1) the scope of copying, and (2) the nature of the copied work. Given the

implications of the idea and the functional considerations, what is the extent of the expression? If there is "essentially only one way to express an idea," complete copying is permissible. *Concrete Mach. Co. v. Classic Lawn Ornaments, Inc.*, 843 F.2d 600, 606 (1st Cir.1988). If there are "only a limited number of ways of expressing the idea," then proof of "near identity" of copied expression is required to show infringement. *Id.* For a work "embodying only one of an infinite variety of ways of expressing an idea," duplication or near identity is not required. *Id.* at 607. Consequently, there is a "sliding scale" that determines the scope of copyright protection. Accordingly, I first examine more closely the scope of copying and then examine how much of this copying was impermissible.

A. Menu Commands and Structure—Scope of Copying.

1. *Defining and describing "menu commands and menu structure."*

Omitting details not relevant for resolution of any dispute in this case, one may describe the idea of menu commands and menu structure in the following way.

Each spreadsheet program described in the evidence before the court has a set of basic executable operations that a user may invoke. Each executable operation does something with data in a spreadsheet (*e.g.*, erases data in a spreadsheet cell), fixes parameters of a spreadsheet (*e.g.*, width of a column in a given spreadsheet), fixes parameters of the program (*e.g.*, hardware configuration or default settings), or performs some other function such as printing or saving a spreadsheet.

For each spreadsheet program before the court, the number of executable operations is large. Also, the possible methods of presenting available operations to the user is large. One simple approach would be to give each operation a unique name. If this were done, however, the large number of unique names would make it difficult for a user to remember and invoke them expeditiously.

A common way of overcoming this problem is to develop a menu hierarchy. The first level of the hierarchy presents the user with a "menu" consisting of a limited number of "menu commands." Some of these menu commands may be executable operations (*e.g.*, "Quit") causing the program to do something (*i.e.*, terminate). When the user invokes an executable command, the program performs the corresponding operation and does not present any further menu options. To proceed farther in the hierarchy, the user must start again and select a different option.

Other menu commands are not in the set of executable operations; instead they are "internal" menu commands, each of which substitutes a new menu in place of the menu in which the internal command appears. Like the menu it replaces, the substituted menu (or "submenu") may consist of a combination of executable commands and internal commands.

The menu commands and menu structure are commonly described as a "tree." The imagery is imperfect, however, unless one thinks of a rather unusual tree that has a leaf or two as well as branches at most junctures. In this imagery, some of the choices at each juncture may be branches and others may be leaves. Each branch or leaf has a name. The "name" of the branch or leaf is a "menu command" within the program's "menu tree". The user starts at the trunk, and by choosing a branch, starts a climb upward. (One may, of course, also envision an upside-down tree, with the user working downward.)

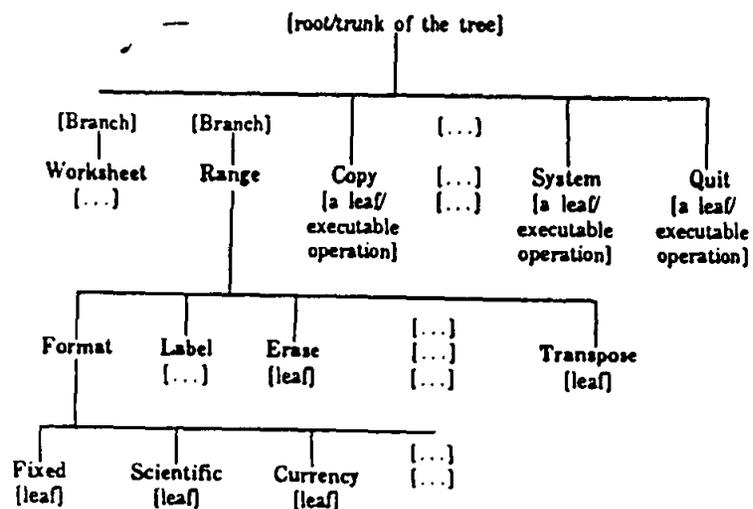
If instead of choosing a branch at the first juncture the user chooses an executable operation, no climb occurs. A name (menu command) corresponding to an executable operation is a leaf of the tree. Having reached that leaf, the user cannot climb farther; no new branches are presented. Instead, the operation is executed. To climb farther (or elsewhere in the tree), the user must go back at least a bit (or all the way to the beginning) and choose branches up to a higher level. As long as the user invokes an internal menu command (a branch) at each juncture, the program presents to the user a new set of

branches and leaves, and by choosing a branch the user climbs higher.

The height of the climb (the number of branches that the user must select before reaching a "leaf" or executable operation) varies depending on which choices the user makes at each juncture. Thus, if "Quit" is selected at the first level, the user makes no climb; the user has selected a leaf, not a branch.

To format a cell as currency data in Lotus 1-2-3, the user climbs three levels up—choosing the range branch, followed by the format branch, followed by the currency leaf.

Switching the imagery to an upside-down tree (a more common convention in computer software discussion), envision a downward progression. For example, the following is a graphic display of the part of the tree a user would climb (or descend) to select the currency operation.



The term "menu tree" refers to the structure of all of the branches and leaves in the menu hierarchy as well as the names of those branches and leaves (*i.e.*, the names of the menu commands).

In the July 31 Memorandum and Order, I concluded that the definition of the "idea," "system," "process," "procedure," or "method" of Lotus 1-2-3 includes the set of executable operations (leaves) of the tree (although not necessarily the menu command name assigned to each operation). As part of the "idea", the determination of the function of each executable operation is not protected by copyright law.

The menu tree expresses to the program user, in hierarchical fashion, the array of available operations. Even though the executable operations are not copyrightable, the menu tree is copyrightable because the (hierarchical) arrangement of the definition and identification of executable operations contains expression. Accordingly, in resolving the parties' motions for summary judgment, I concluded that many of the details of the menu tree in Lotus 1-2-3 were not a part of the "idea," "system," "process," "procedure," or "method" of the Lotus 1-2-3 program, but rather a part of the expression of that idea.

2. The Lotus and Borland menu tree.

The original complaint alleges infringement of Lotus's copyright in Lotus 1-2-3 version 2.0 (and earlier versions). In deciding the issues before the court, I have examined both Lotus 1-2-3 version 2.0 and version 2.01. Version 2.01 is a copyrighted derivative work based on version 2.0. Borland points to no differences between these programs that would have a bearing on any issue in this case. A comparison of the menu tree for Lotus version 2.0 (Exhibit 524) and the menu tree accessible through the version 2.01 program also demonstrates no differences material to this litigation. Indeed, the parties earlier stipulated (1) that the menu commands are the same and are displayed in an identical fashion to the user, and (2) that no difference exists between the user interfaces that would have any effect on the issues in this case. See O'Connor Dec'n. Docket No. 108, Exh. H. Accordingly, I will

refer to the menu tree generically as the Lotus 1-2-3 menu tree although I have examined both versions 2.0 and 2.01.

The Lotus 1-2-3 menu tree contains approximately 469 menu commands. See Kieras Dec'n, V:& at ¶ 66.

(The record in this case consists of nine volumes of material. Citations to this record are formatted as Volume:Tab; thus "V:8" means Volume V, Tab 8.)

Portions of the menu tree are illustrated at various places in the Exhibits. The entire tree is set out in Exhibit 524, A-C, E (the "Flesher Exhibits" which were originally submitted with the summary judgment materials). As the number of commands suggests, the menu tree is quite large.

The Quattro and Quattro Pro programs have both a native menu tree and a "1-2-3 emulation" menu tree. I do not understand Lotus to contend that the native menu tree infringes any Lotus copyright. Accordingly, when I refer to the Quattro or Quattro Pro menu trees, I refer only to the menu tree used in the program's 1-2-3 emulation interface.

The Quattro menu tree contains approximately 3370 menu commands. (Kieras Dec'n, V:8 at ¶ 77.) This menu tree is illustrated in Flesher Exhibit B. The Quattro Pro version 1.0 menu tree contains approximately 5216 commands (*Id.* at ¶ 66.) This menu tree is illustrated in Flesher Exhibit A.

The Quattro Pro program has been released in several versions, specifically versions 1.0, 2.0, 3.0, 4.0, 4.01, SE, and Quattro Pro for Windows. Only versions 1.0, 2.0, 3.0, and 4.0 contain the emulation interface. In deciding issues for Phase I of the trial, therefore, when I refer, without further specification, to the Quattro Pro menu tree, I refer only to the emulation menu tree in Quattro Pro versions 1.0, 2.0, 3.0, and 4.0.

The entire Lotus menu tree is contained within both the Quattro and Quattro Pro version 1.0 menu trees. For example, the first menu in Quattro Pro version 1.0 contains an identical copy of the 1-2-3 menu commands, in the same order, but with one new command inserted ("View"). Invoking a Borland command that is identical to a Lotus command produces a menu that is an identical copy of the Lotus submenu, but (in

some cases) with one or two new commands inserted. Thus, each menu or submenu in Lotus 1-2-3 is reproduced identically, but with the insertion in the Quattro and Quattro Pro menu trees of some new menu commands and any submenus associated with the new menu commands.

Put another way, both the Quattro and Quattro Pro version 1.0 menu trees consist of a *virtually identical* copy of the entire 1-2-3 menu tree, with new branches or leaves inserted at various places.

Although the above comparison focuses only on version 1.0 of the Quattro Pro menu tree, the scope of copying remains the same for versions 2.0, 3.0, and 4.0. The only difference among the emulation menu trees in Quattro Pro versions 1.0, 2.0, 3.0, and 4.0, is that each successive version added new commands. See Warfield Dep'n, III:40 at 72-73. Thus, later versions also contain a virtually identical copy of the Lotus menu tree, with still more new branches or leaves added.

The presentation of the commands in the Quattro and Quattro Pro programs indicates which commands are found in the 1-2-3 menu tree and which are Borland's insertions. In Quattro, the new (and only the new) menu commands are followed by a dot at the place where the new command is inserted into the tree. Thus, in Quattro, the "Install" command was inserted into the first level menu followed by a dot. The submenu of "Install" does not have these same dot designations. Such a designation is unnecessary for this submenu, however, because everything within the newly inserted "Install" branch of the tree is new.

Quattro Pro uses the same approach for differentiating 1-2-3 menu commands and added menu commands, with the exception of "View" in Quattro Pro's first menu, which has no designation as an inserted command but whose submenu consists of commands each followed by a dot. In addition, in the Quattro Pro menus, a line generally appears separating added menu commands from original 1-2-3 menu commands.

The disparity in total number of menu commands between the 1-2-3 menu tree and the Quattro and Quattro Pro menu trees is large, but does not alter the scope of copying. Most of

the new commands are clustered together far up in the tree (or down in the upside down tree). For those menus that exist in the Lotus 1-2-3 menu tree, the number of added menu commands is rarely more than one or two. The effect is similar to an identical copy of a book with some paragraphs and lengthy footnotes inserted, and some voluminous appendices attached at the end.

In short, I adhere to my earlier conclusion that no reasonable factfinder could find that Borland did not take from Lotus 1-2-3 the menu commands and menu command structure substantially as they were. Further, I find that Borland produced a virtually identical copy of 1-2-3 menu tree, albeit with additions, in its Quattro and Quattro Pro emulation interfaces.

B. Nature of the Work—Scope of Illicit Copying.

1. *Scope of limitations imposed by functional considerations.*

a. *The functional considerations.*

Borland advanced the following "constraints" on design of a menu tree.

- (1) Each menu command was chosen to tell the user the purpose of the menu command and its function.
- (2) Each menu command was selected so that it had a different first letter from the other menu commands within the same menu.
- (3) Each menu was set up to have only about seven choices (witnesses referred to this as the "seven plus or minus two rule," *i.e.*, no menu should have fewer than five or more than nine commands).
- (4) The menus were structured so that words within the menu that dealt with similar functions were grouped together.

- (5) Executable operations that were likely to be frequently used were located near the top of the (upside down) tree.
- (6) Menu commands within a menu were arranged from left to right in order of decreasing frequency of use.
- (7) Commands in submenus were grouped under the menu command to which they relate.

See VII:A:6. Borland also proffered an eighth functional "constraint" of having each menu fit on one line of a computer screen. If this constraint were followed, each menu could have no more than 80 characters (the number of characters across a standard computer terminal screen).

Borland refers to the above items as "constraints" on design of a menu tree. Lotus contends that they are "guidelines" or "rules of thumb." The difference in terminology is largely if not entirely semantic. Nevertheless, the different terms raise different images. "Constraint" implies a rule that must not be violated. If "constraint" is defined this way, the eight listed items are not "constraints." Each of them is violated somewhere in the Lotus 1-2-3 menu tree (except possibly the second, which has been violated in other menu trees, *e.g.*, Excel 3.0 with "Save" and "Save As" in the same menu, and the last, which was violated in SuperCalc). Moreover, the alleged "constraints" are full of terms that offer little guidance on how to conform. For example, when is one menu command "similar" to another? The answer may be easy in some cases and entirely within the discretion of the programmer in others.

"Guideline," or "rule of thumb," is a more accurate description. Each of these terms implies a directive that is not always to be followed. Even these terms, however, ordinarily imply a precise rule. The first item listed above is hardly precise, even though it does limit the possible forms of a menu tree. Accordingly, I will refer to these as functional "considerations."

b. *Impact of functional considerations on form of the menu tree.*

The parties dispute the impact of the eight functional considerations (listed above) on the freedom of expression in forming a menu tree. For the following reasons I conclude that the Lotus 1-2-3 menu tree is just one of a great variety of possible expressions that are consistent with the functional considerations listed above and the specific set of executable operations used in Lotus 1-2-3.

First, none of the proffered functional considerations is overriding or dictates any specific result. Each may be violated. Most are violated at one or more points within the Lotus 1-2-3 menu tree. Indeed, the considerations are often competing and must be traded off against each other. *See, e.g.,* Gottheil Dep'n, IX:11 at 143-64; Olson Dep'n, III:32 at 117-18. No functional considerations are offered for guiding the determination of how to trade off competing concerns. Borland provided no credible evidence explaining how functional considerations could completely control formation of the menu tree.

Second, the "quality" of the menu tree depends on the peculiarities (*i.e.*, the particular tastes) of the individual using the program. *See, e.g.,* Bosworth Dep'n, I:1 at 389-90. For this to be true, there must be a variety of possible menu trees for the user to choose from. In this regard, Borland programs offer a "menu builder" that enables users to alter, customize, and create menu trees. If functional considerations restricted the possible expression of the menu tree to a limited number of possibilities, there would be little or no need for a user to modify it.

Third, even Borland's experts acknowledge that, given all of the various functional considerations, at least a limited range of choices remains for individual menu commands. *See* Liddle Dep'n, II:27 at 121; Olson Dec'n, V:13 at ¶ 35. Thus, "Copy" could be called "Replicate," "Duplicate," "Reproduce," "Repeat," "Ditto," etc. "Range" could be "Block," "Scope," "Extent," "Cells," etc. Although some experts contend that certain words and menu structure are preferable to

others, these same experts contend that the words Lotus selected did not matter for 1-2-3's success. *See* Liddle Dec'n, V:11, ¶ 55; Olson Dec'n, V:13, ¶¶ 36, 38. In any event, even if there were as few as two acceptable words for each menu command (given the other functional considerations), there would be 2 raised to the 469th power possible menu trees (an astronomical number) having precisely the same menu structure as the 1-2-3 menu tree, but with variations in menu command names. *See* Emery Dec'n, V:3, ¶ 85. Even if only half of the menu commands had more than one possible name (given the other functional considerations), there are over 2²³⁴ possible menu trees having the same menu structure.

In listing these examples, I do not mean to suggest that the alternatives for menu command names are so few, nor to imply that every synonym for a command word is suitable in view of the other functional considerations. Rather, I merely note the breadth of possible menu trees that may be achieved in this manner as a factor bearing upon whether implementation of the idea, system, process, procedure, or method underlying the Lotus 1-2-3 menu tree is capable of a wide variety of expression.

Also, in presenting this analysis, I do not imply that counting possible variations in individual words is decisive of the breadth of possible expressions for literary works in general. It would not necessarily be significant, for example, to determine how many ways instructions for a simple game can be expressed solely by examining possible alternatives for individual words. Nevertheless, variations of the words in the menu tree represent a material consideration in the context of this case because of (1) the nature of the menu tree, which unlike an English sentence, permits substitution of individual words without changing the meaning of the menu tree, (2) the size of the menu tree, and (3) the structure of the menu tree.

If the designer of a menu tree chooses not to copy the structure of Lotus 1-2-3, the designer's freedom of expression and range of possible expressions for the menu tree expand dramatically. *See* Emery Dec'n, V:3, ¶ 84. Nothing in the materials before me supports an inference that functional considerations alone control the structure of the menu tree.

Indeed, Borland's experts implicitly acknowledge alternatives for the structure of the 1-2-3 menu tree. See Liddle Dec'n, V:11, ¶ 56; Olson Dec'n, V:13, ¶ 38.

Finally, a number of other spreadsheet programs use vastly different menu trees. The existence of vastly different menu trees in other commercial programs supports the conclusion that the Lotus 1-2-3 menu tree is but one of many possible forms for a menu tree.

Borland contends, however, that the differences in menu trees between programs such as Lotus 1-2-3 and Excel are due to differences in the programs' functionality. That is, Borland argues that because the executable operations (leaves of the tree) are different and the visual displays are different (*e.g.*, pop-up menus vs. text on a single line), the menu trees are different. Borland contends that, as a result, the existence of other commercial programs with menu trees vastly different from the 1-2-3 menu tree (and each other) is not probative of the degree of freedom of expression that exists in formulating the 1-2-3 menu tree.

I reject Borland's contentions for the following reasons. Although differences in program functionality may explain some differences between menu trees, the differences in functionality cannot explain the breadth of differences among menu trees used in the various programs. For example, functional considerations do not explain why the "File" menu command is left-most in the Excel menu tree and fifth from the left in the 1-2-3 menu tree. Borland does not suggest or offer any evidence explaining how the (unspecified) differences in functionality among 1-2-3, Excel, and Quattro's native mode affect placement of the "File" command within the first menu. Similarly, 1-2-3 places the copy and move command in the first level of the menu tree. In Excel they are in the second level.

The differences among menu trees in the various programs submitted are so large that they are, in a practical sense if not literally, incapable of enumeration. The broad scope of these differences cannot be explained in terms of differences in functionality. Indeed, Borland offers no evidence or argument providing a reasoned explanation (as opposed to an un-

ported assertion) of how the magnitude of differences could be explained by any differences in functionality. I conclude that many of the differences are due to different choices about how to express to the user the available user choices about all the particular operations that the program can perform.

In argument, Borland placed primary emphasis on an expert's assertion that the set of possible menu command names is "not large." See Olson Dep'n, III:32 at 54-59. As a preliminary matter, I find that this testimony is entitled to little, if any, weight. Nothing purports to explain how the expert reached this conclusion. In fact, this expert admits that she has never attempted to determine how large the set is. An expert's unexplained and unsupported assertion, even if received into evidence, is scant basis for reaching a reasoned finding.

In addition, the testimony is unclear whether the expert meant that there was a small set of choices for individual command names, or the set of all menu command names. In an earlier declaration this expert conceded that there is a "very narrow range" of suitable words for individual commands. See Olson Dec'n, V:13 at ¶¶ 35-36. As observed above, even if (given all the functional considerations) only half the commands have only one alternative label, the possible number of menu trees that differ in detail remains very high. Consequently, what this expert meant by "not a large set" may be entirely consistent with a determination that the number of possible ways of expressing the menu tree is without limitations that are material to the separation between idea or function and expression.

Finally, even assuming that the alternatives for menu command names were few, when I take into account as well the options available for structure, I find that the number of possible choices again expands dramatically. Thus, what Borland copied from 1-2-3 was not limited to aspects dictated by functional considerations. Rather, Borland copied the entire menu tree, much of which was the free expression of the creators of Lotus 1-2-3.

c. Conclusion

As Borland practically conceded in closing argument, application of functional considerations does not restrict the expression of the menu tree to essentially only one form. (See Docket 333 at 3-52.) Further, I find that although functional considerations may have some effect on the design of a menu tree, they do not impose any practical limitation on the possible forms of expression to a number far enough short of infinity that any way of expressing the number in English words has come into common usage. The set of executable operations in Lotus 1-2-3 is large and the possible structural variations are enormous. The menu tree is capable of a very wide variety of expressions.

Borland has not argued or provided any evidence that any specific aspect of menu structure or command names, short of the entire menu tree, is dictated solely or influenced mainly by functional considerations. Moreover, Borland has used a virtually identical copy of the 1-2-3 menu tree. Accordingly, this case on its facts does not present any issue that might arise on a finding of copying of something short of virtually the entire menu tree.

2. Originality.

In its closing argument, Borland asserted that, because of functional considerations, creation of the menu tree did not require sufficient originality to justify protection under copyright law.

a. Originality doctrine.

The Supreme Court recently addressed the requirement of originality in *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 111 S.Ct. 1282, 113 L.Ed.2d 358 (1991). The Opinion for the Court in *Feist* observed that the requirement that a work be "original" derives from the Constitution. The Court described the originality requirement as follows.

The *sine qua non* of copyright is originality. To qualify for copyright protection, a work must be original to the

author. Original, as the term is used in copyright, means only that the work was independently created by the author (as opposed to copied from other works), and that it possesses at least some minimal degree of creativity. To be sure, the requisite level of creativity is extremely low; even a slight amount will suffice. The vast majority of works make the grade quite easily, as they possess some creative spark, no matter how crude, humble or obvious it might be.

Id., 499 U.S. at ___, 111 S.Ct. at 1287 (internal quotation marks and citations omitted).

The Court held that copying from the white pages of a telephone book was permissible. To reach this conclusion, the Court first confirmed the firmly established principle that facts (*e.g.*, the names, towns and telephone numbers in the white pages) are not copyrightable. The Court also confirmed that "even a directory that contains absolutely no protectable written expression, only facts, meets the constitutional minimum for copyright protection if it features an original selection or arrangement." *Id.*, 499 U.S. at ___, 111 S.Ct. at 1289. In this circumstance, "the copyright is limited to the particular selection or arrangement." *Id.*, 499 U.S. at ___, 111 S.Ct. at 1290.

Applying these two basic principles, the Court determined that the selection and arrangement of facts in the plaintiff's telephone book, *i.e.*, selection of all people who applied for phone service and arrangement in alphabetical order, was "devoid of even the slightest trace of creativity." *Id.*, 499 U.S. at ___, 111 S.Ct. at 1296.

[O]riginality is not a stringent standard; it does not require that facts be presented in an innovative and surprising way. It is equally true, however, that the selection or arrangement of facts cannot be so mechanical or routine as to require no creativity whatsoever.

Id. Indeed, the selection of all people and placement in alphabetical order was not at all original to the plaintiff—phone books had been arranged in that fashion for years. Plaintiffs

white pages were nothing but a "garden-variety" phone book. *Id.*

b. Originality in this case.

Relying on *Feist*, Borland contends that the eight "rules" regarding functionality listed above demonstrate that the form of the menu tree (including menu command names and structure) is not original. For the following reason, I find that the menu tree is "original."

As explained above, the menu tree of Lotus 1-2-3 may be viewed as an arrangement of the executable operations as "leaves" of the "menu tree," with internal commands ("branches") leading through the structure to the "leaves." Altogether, the tree contains 469 labels to explain the arrangement. This text and arrangement presents the set of executable operations to the program user. Thus, the specific operations could be viewed as (uncopyrightable) facts and the menu tree as an arrangement (with textual labels) of the facts. Even viewed in this light, most favorably to Borland, the menu tree easily satisfies the originality requirement discussed in *Feist*.

The functional considerations Borland propounds are not at all comparable to application of a rule of alphabetical ordering of executable operations.

Borland's "rules" are violated with regularity in the 1-2-3 menu tree. In contrast, the rule of alphabetical ordering was not violated in the *Feist* white pages.

Also, the functional considerations listed above were not "an age old practice, firmly rooted in tradition and common place." *Feist*, 499 U.S. at ___, 111 S.Ct. at 1297.

Moreover (except in a few places where rules of alphabetical ordering differ) one who is given the names, addresses and numbers in a phone book, can generate the alphabetical ordering by routine and mechanical application of rules, with only one possible outcome. In contrast, the text for the menu commands and the menu structure itself are not dictated by mechanical application of the functional considerations. As noted above, a wide variety of menu trees is possible.

In sum, the 1-2-3 menu tree "make[s] the grade quite easily, as it possess[es] some creative spark. . . ." *Feist*, 499 U.S. ___, at 111 S.Ct. at 1287.

Borland nevertheless contends that, because functional considerations played a role in formulating the 1-2-3 menu tree, the menu tree is not copyrightable. Borland's argument is susceptible of two interpretations, both lacking merit.

First, one may interpret this argument as a contention that the functional considerations so permeate formulation of the menu tree that the menu tree is not separable from the "idea" of the program. This form of argument has nothing to do with the amount of creativity or originality involved. Derivation of a scientific formula may require a great deal of creativity and produce an original result. If the formula fails the copyrightability test, it is because the formula is not expressive—independently of creativity or originality. Casting the argument in terms of originality doctrine rather than separability does nothing to assist resolution of the issues in this case, and may lead to confusion. In any event, to the extent Borland raises the same separability argument that it has raised before, but dressed now in terms of originality, that argument was previously rejected by the court for good reason, and is now rejected again.

Alternatively, Borland may be understood as contending that any work whose form is restricted to any material extent by functional considerations is not original. Without more, this contention is invalid on its face. Any original literary work is formulated according to functional considerations imposed by language, a desire for clarity, and a desire to express the ideas conveyed. As one of Borland's experts concedes, the first functional consideration (conveying the nature of executable operations) is not materially different from a functional consideration for selection of words in any English writing. See *Olson Dep'n*, III:32 at 54. Accordingly, this type of "functional consideration" can remove a writing from copyright protection only if it restricts the forms of expression (that are separable from the idea or function of the work) to a limited number. Thus, cases referring to functional con-

siderations and decided under originality doctrine, including those cited by Borland, uniformly refer to limits on the number of forms of expression given functional considerations.

See Victor Lalli Enter., Inc. v. Big Red Apple, Inc., 936 F.2d 671, 673 (2d Cir.1991) ("purely functional grids that offer no opportunity for variation"); *Sinai v. California Bureau of Automotive Repair*, 25 U.S.P.Q.2d 1809 [1992 WL 470699] (N.D.Cal.1992) ("limited number of ways" to arrange information in chart); *New Haven Copper Co. v. Eveready Mach. Co.*, 229 U.S.P.Q. 838 [1986 WL 269] (D.Conn.1986) (column headings "dictated by functional considerations"); *Merritt Forbes & Co. v. Newman Inv. Sec., Inc.*, 604 F.Supp. 943, 951-52 (S.D.N.Y.1985) (no originality if "form of expression is dictated solely by functional considerations"; determining that fact issue exists over whether concept is "capable of varied expressions"); *Decorative Aides Corp. v. Staple Sewing Aides Corp.*, 497 F.Supp. 154, 157 (S.D.N.Y.1980) (what was similar was "dictated by functional considerations"), *aff'd without op.*, 657 F.2d 262 (2d Cir.1981). *Cf. Feist*, [499 U.S. at ___], 111 S.Ct. at 1296 (routine and mechanical application of single rule).

Thus, Borland's argument fails because, as above, a wide variety of expression is possible for the Lotus 1-2-3 menu tree.

C. Value of Expert Testimony.

In reaching the conclusions above, I have read and weighed the expert testimony offered by the parties. For the following reasons, however, much of this testimony has little weight in relation to the issues of law and fact that are decisive of the outcome in this case.

First, most of the testimony expresses conclusions, without any reasoned explanation of the basis for the conclusions. Second, much of the testimony uses terms such as "system" or

"a large number" that the experts do not define. As factfinder, I choose not to abandon my factfinding responsibility by accepting an expert's proposed unexplained choice. This is my usual practice, and for even stronger reasons I proceed in this way when the expert's testimony uses terms specially defined by the expert to have meanings that incorporate into the terminology substantive choices that have important policy implications, without examining and explaining reasons for the policy choices and whether they are consistent with applicable statutes and precedents.

Finally, much of the testimony is based on explicit or implicit assumptions about copyright law that are incorrect. For example, one expert testified that she was unsure of whether design of a menu tree is creative. In defining creativity, however, the expert explained that creativity required some "inventive leap." Olson Dec'n, III:32 at 124-35. Of course, this type of creativity—the inventive leap or new idea—is not required for copyrightable expression. *See Feist*, 499 U.S. at ___, 111 S.Ct. at 1296 (novelty or innovation not required). Expression is copyrightable, even when it is a new expression (e.g., a new novel about young love) of an old idea (i.e., boy meets girl).

D. Conclusion.

In sum, I conclude that each of the Borland emulation interfaces contains a virtually identical copy of the 1-2-3 menu tree and that the 1-2-3 menu tree is capable of a wide variety of expression.

III. Affirmative Defenses for Phase I.

A. Laches.

Borland has the burden of proving laches. *Costello v. United States*, 365 U.S. 265, 282, 81 S.Ct. 534, 543, 5 L.Ed.2d 551 (1961). A defense of laches has been established if the defendant proves that (1) the plaintiff inexcusably or

unreasonably delayed in bringing an infringement action, and (2) defendant was prejudiced by this delay. *See id.* *See also* 3 Nimmer on Copyright § 12.06.

The chronology of the principal events bearing upon Borland's defense of laches commences in January 1987, when Lotus filed suit in the *Paperback* case. Borland was developing Quattro at the time. Before including the 1-2-3 emulation interface in Quattro, Borland secured a legal opinion that the program did not infringe Lotus's copyright. In late September 1987, Borland announced Quattro, Borland's first spreadsheet product. In November, Borland shipped the Quattro product. Quattro was advertised widely, one advertisement appearing in a November 1987 issue of "Lotus Magazine."

In early 1988, the Quattro product was discussed at a high-level Lotus meeting. The evidence concerning this meeting supports an inference that certain Lotus executives discussed the possibility that Borland would change Quattro if Lotus threatened suit. The evidence does not prove, however, that Lotus delayed filing suit or kept silent on the issue of infringement *for this reason*. I find that Lotus may have discussed the possibility that Borland would change its product, but that this was not a factor in Lotus's decision not to file suit in 1988.

Some time after the meeting, Lotus decided not to file suit against Borland then. In addition, Lotus adopted a policy of not commenting on possible infringement by, or legal action against, Borland.

In September 1988, less than one year after Borland shipped Quattro, Borland acquired the "Surpass" spreadsheet technology (from "Surpass Software Systems") for approximately \$2.4 million. This included a software program that had been called "Surpass." The Surpass program's only menu tree was a copy of the 1-2-3 menu tree. After acquiring Surpass, Borland removed the Surpass product from the market. Borland then re-engineered Surpass to become the Quattro Pro product, having a 1-2-3 compatible menu tree as just one of the possible menu trees available. Quattro Pro version 1.0 was released in November 1989.

Near the end of June 1990, this court handed down its decision in the *Paperback* case. The day after the decision was released, Borland filed an action in California seeking a declaration of noninfringement. On July 2, 1990, Lotus filed the original complaint in the present action.

1. Delay.

The period between Borland's release of Quattro and Lotus's filing of the instant action was approximately 2½ years. This amount of time, standing alone, does not dictate a finding that the delay was unreasonable.

Cf. Roulo v. Russ Berrie & Co., 886 F.2d 931, 942 (7th Cir.1989) ("A two year delay in filing an action following knowledge of the infringement has rarely been held sufficient to constitute laches."), *cert. denied*, 493 U.S. 1075 [110 S.Ct. 1124, 107 L.Ed.2d 1030] (1990); *Hoste v. Radio Corp. of America*, 654 F.2d 11 (6th Cir.1981) (per curiam) (error to grant summary judgment based on laches for suit filed after 13 year delay); *Boothroyd Dewhurst, Inc. v. Poli*, 783 F.Supp. 670, 680 (D.Mass 1991) (four year delay may not constitute laches).

Whether this amount of delay was unreasonable or inexcusable depends on the motives for delay. Thus, Judge Learned Hand observed that

it is inequitable for the owner of a copyright, with full notice of an intended infringement, to stand inactive while the proposed infringer spends large sums of money in its exploitation, and to intervene only when his speculation has proved a success. Delay under such circumstances allows the owner to speculate without risk with the other's money; he cannot possibly lose, and he may win.

Haas v. Feist, 234 F. 105, 108 (S.D.N.Y. 1916). On the other hand, delay used to evaluate and prepare a complicated claim may be reasonable. *See, e.g., Paperback*, 740 F.Supp. at 82.

Similarly, delay to determine whether the scope of proposed infringement will justify the cost of litigation may be permissible. *See, e.g., Boothroyd Dewhurst*, 783 F.Supp. at 680-81. In the circumstances of the present case, I find that Lotus's delay in filing suit was reasonable and excusable.

First, I reject Borland's contention that Lotus's delay in bringing suit is of the kind addressed in *Haas*, *i.e.*, in order to speculate with Borland's money. Borland relies primarily on the failure of Lotus to file an action in the early part of 1988. Borland contends that Lotus then knew enough about Borland's plans that Lotus should have notified Borland immediately if Lotus intended to claim infringement and that Lotus did not do so because Lotus was afraid that Borland would change Borland's product. Borland contends, in other words, that Lotus waited to file suit in an effort to trap Borland into expending large sums of money on Borland's spreadsheet products and, later, to reap the benefits of Borland's investment.

I find, however, that the evidence Borland presented does not support Borland's contentions. At the time of the 1988 meeting, Borland had already developed, advertised, and shipped Quattro. The possibility that Borland would adjust Quattro to remove the copy of the Lotus 1-2-3 menu tree was marginal in significance at the time of this meeting. For this reason, reinforced by the findings stated above, I do not find the evidence sufficient to persuade me, as factfinder, to draw the inference that Lotus waited to sue hoping that its competitor's product would succeed and enable Lotus to reap enhanced damages.

See Russell v. Price, 612 F.2d 1123, 1126 (9th Cir.1979) (conclusion that plaintiff was not speculating on defendant's success supported by fact that plaintiff was also seeking to exploit the copyrighted material), *cert. denied sub nom., Drebin v. Russell*, 446 U.S. 952 [100 S.Ct. 2919, 64 L.Ed.2d 809] (1980).

I find that Lotus delayed bringing suit for the purpose of awaiting resolution of the *Paperback* and *Mosaic* cases.

Throughout 1988, Lotus was involved in extensive litigation in *Paperback* and *Mosaic*. This litigation was hotly contested. If Lotus had lost the *Paperback* case, Lotus's claims against Borland would have been difficult to maintain. Thus, by waiting to file suit, Lotus avoided expensive duplicative litigation all of which might have been unsuccessful if Lotus lost in *Paperback*. This avoided a risk of needlessly wasting court and party resources. Once the *Paperback* case was resolved by the trial court, Lotus filed suit against Borland almost immediately. I cannot say that this was unreasonable.

Borland contends, nevertheless, that Lotus's delay was unreasonable in part because Lotus kept silent about its belief that Borland's product infringed. Lotus responds that it was satisfied that Borland knew of the infringement claims because of (1) Phillip Kahn's (Borland's chairman's) strong public reaction condemning Lotus for filing the *Paperback* case, and (2) a public statement Lotus issued before Quattro was shipped that appeared in the May 26, 1987 issue of PC Magazine (Exh. 28 at 162-63). In the public statement, Lotus represented that it did not claim copyright over the ideas of a two-line moving cursor interface or context sensitive help. Lotus then characterized its suit against Paperback and Mosaic as protecting against companies that copied "all of the ways in which 1-2-3 communicates to the user, including its menu structure and sequence, word selection, and macro language design. . . ."

I find that Lotus reasonably believed that Borland was aware of Lotus's potential claims against Quattro and Quattro Pro. Borland "was far too sophisticated to need being led by the hand." *Famous Music Corp. v. Bay State Harness Horse Racing & Breeding Assoc., Inc.*, 554 F.2d 1213, 1215 (1st Cir.1977) (rejecting estoppel defense). In light of this, I cannot say that Lotus's decision not to comment on the infringing nature of Quattro or Quattro Pro makes Lotus's delay in this case unreasonable or inexcusable.

In weighing equities, some courts decline to permit a laches defense by a "deliberate pirate," *Haas*, 234 F. at 108, or less harshly stated, a "deliberate infringer."

See 3 Nimmer on Copyrights, § 12.06 at 12-107 (delay "may not be a bar against one who knew of plaintiffs asserted rights, or as against a deliberate infringer") (footnotes omitted).

I find that Borland (1) knew of Lotus's copyright, (2) knew that Lotus was acting to protect that copyright, and (3) deliberately introduced a virtually identical copy of the Lotus 1-2-3 menu tree into its 1-2-3 emulation interfaces.

The fact that Borland did secure a noninfringement opinion from a prestigious law firm is relevant, of course. Nevertheless, Lotus is not in any way accountable for that opinion. Indeed, Borland offers no evidence that Borland gave any notice to Lotus of Borland's seeking legal advice or of the substance of that advice. It is a rather curious twist of argument to suggest that a copyright owner has a duty of notice to the infringer when the infringer has enough concern about its own actions that it seeks a legal opinion privately but refrains from giving any kind of notice to the copyright owner.

Borland contends that the *Paperback* suit could not constitute notice that Lotus was enforcing its copyright because the issues in *Paperback* and this case are now identical. The fact that the issues are not identical, however, is far from decisive. More to the point is the fact that Borland was aware of Lotus's assertion of copyright protection for the 1-2-3 program and in particular for the form of its menu tree.

Borland next argues that Lotus had not sufficiently formulated its precise contentions of what aspects of Borland's products infringe Lotus's copyright until May of 1992—almost two years into this case. Again, this has little to do with the fact that Borland was aware of Lotus's assertion of copyright protection for Lotus 1-2-3 and in particular for the form of its menu tree.

2. Prejudice.

The defense of laches requires that defendant suffer prejudice. Typical forms of prejudice include: death or unavailability of an important witness, dulling of memories, loss of relevant records, and continuing investments and outlays by

the alleged infringer in connection with the operation of its business. See *Eisenman Chemical Co. v. NL Indus., Inc.*, 595 F.Supp. 141, 147 (D.Nev.1984). Borland argues only the last form of prejudice. To demonstrate prejudice, Borland points to marketing costs and acquisition of the Surpass technology.

To constitute prejudice, however, these expenses must have been incurred *as a result* of Lotus's delay in bringing suit. Where an infringer was aware of a plaintiff's copyright, as Borland indisputably was, courts have phrased this point as a requirement that the actions be taken in reliance on the plaintiff's delay in bringing suit.

See *Russell*, 612 F.2d at 1126; *In Design v. Lauren Knitwear Corp.*, 782 F.Supp. 824, 831 (S.D.N.Y.1991); *Steinberg v. Columbia Pictures Indus., Inc.*, 663 F.Supp. 706, 716 (S.D.N.Y.1987).

For the following reasons, I conclude that Borland has not proved that it took any action in reliance on Lotus's delay in bringing suit.

First, I find that Borland would have invested in and entered the spreadsheet market whether or not Lotus delayed in bringing suit. Phillippe Kahn testified that the decision to include the 1-2-3 emulation interface in Quattro was made not long before the product shipped. (I:14 at 362). Of course, this initial decision to include the 1-2-3 emulation interface was made before Lotus could have objected.

The Surpass technology, acquired in 1988, was used to form the basis for Quattro Pro. This was less than one year after Borland shipped Quattro. Borland presented no evidence that it would not have acquired Surpass had Lotus taken action earlier. Moreover, Quattro already had a 1-2-3 emulation interface. It would not be reasonable to infer that other aspects of Surpass than its copy of the Lotus 1-2-3 menu tree had no influence on Borland's decision. I find that Borland has not proved that it acquired the Surpass technology in reliance on Lotus's delay in bringing suit.

I observe also that Borland filed a declaratory judgment action immediately after the decision in the *Paperback* case.

Borland seeks to explain the timing of the filing of the declaratory judgment action by presenting evidence that Borland was responding to recent conflicting reports that Lotus would file suit against Borland. Even if I credited this self-serving assertion, unsupported by any objective evidence, however, I would not infer that Borland had ever relied on Lotus's failure to assert infringement to conclude that Lotus would never file suit. Moreover, a senior Borland official averred that the Borland Board of Directors had been informed before the decision in the *Paperback* case that Lotus intended to sue Borland if Lotus prevailed in *Paperback*. (Leyton Dec'n, VI:10, ¶ 5.) Nevertheless Borland, like Lotus, awaited the outcome of the *Paperback* case. Significantly, when Borland filed, it sought a declaration of noninfringement, but omitted any specific request for declaration of unenforceability due to laches or estoppel.

Based on the evidence before me, I find that Borland developed and marketed Quattro and Quattro Pro in reliance on (1) the noninfringement opinion, and (2) the hope that Lotus would lose the *Paperback* litigation. Further, I find that Borland has not proved that it took any actions in reliance on Lotus's delay in bringing suit. Finally, I find that reliance, even if it had occurred, would have been unreasonable in the circumstances shown by the evidence in this case.

B. Estoppel.

The parties apparently agree that, to establish a defense of estoppel, Borland must prove that Lotus engaged in (1) conduct that induced Borland to change its position in good faith, or (2) conduct on which a reasonable person would rely. See *Precious Metals Associates, Inc. v. Commodity Futures Trading Comm'n*, 620 F.2d 900, 908 (1st Cir.1980). In view of this agreement, I need not and do not consider whether this interpretation of *Precious Metals* and other precedents is more generous to Borland's estoppel claim than a more precise understanding of the precedents would be.

To establish estoppel, Borland relies primarily on Lotus's delay in bringing suit and Lotus's failure to inform Borland of Lotus's belief that the Borland products infringe. For the reasons explained in Section A above, I find that (1) Lotus did not intend that Borland infer from Lotus's silence that copying of the menu tree was permissible, (2) Borland was aware of Lotus's copyright, (3) Borland did not rely on the silence of Lotus to Borland's detriment, and (4) any reliance by Borland in the circumstances of this case would have been unreasonable and unjustifiable.

Borland also offers a collection of actions or statements by Lotus that Borland contends entitled Borland to infer that Lotus would not sue Borland for copyright infringement. Borland cites a January 1987 InfoWorld article, released before Lotus had any reason to know of Borland's Quattro product. The article purports to quote a Lotus spokesman, Greg Jarboe:

"Some folks have misinterpreted that what we have done is to copyright the spreadsheet or the two-line [command] interface, neither of which we are trying to do," Jarboe said. The suit was targeted at two vendors who had "copied 99 percent" of a Lotus product, he said, adding that the company is considering issuing a position statement to clarify the limits of its copyright.

InfoWorld, January 26, 1987. Even assuming this is an accurate quote, something Mr. Jarboe denies, I find that it would not be reasonable to launch a spreadsheet product in reliance on this statement. First, as Borland's general counsel concedes, II:20 at 272, the statement does not purport to disavow filing suit against future products that infringe Lotus's copyright in a manner different from the *Paperback* suit. A characterization of the *Paperback* case as involving "clones" is not reasonably interpreted as expressing an intention not to sue other infringers as well as makers of "clones." Second, the article refers to an upcoming position statement (the PC Magazine statement quoted in Section III.A, above). Although Borland contends that none of Borland's responsible representatives saw this position statement (a contention that I find

questionable in light of their other activities), their failure to see and take account of that position statement, along with acting on the assumption that no such statement had been made, was unreasonable. Thus, to the extent Borland claims that it relied on the InfoWorld statement, it was unreasonable for Borland to rely on that statement while failing to take further steps to determine what Lotus's position statement contained.

Borland next cites the fact that Lotus Magazine published advertisements for Borland's products. I find that it would not be reasonable for Borland to assume that the acceptance of the advertisement in Lotus Magazine showed that Lotus had determined that it would make no infringement claim against Borland. Following Borland's line of reasoning, one would conclude that by allowing competitors to run advertisements in Lotus Magazine. Lotus Development Corporation agreed that the competitor's advertising claims were correct (e.g., that Borland's products are superior, as the advertisements implied) and that Lotus wanted individuals to purchase from Lotus's competitors. Merely stating this chain of inference is enough to expose its unreasonableness. I find, instead, that the fact that Lotus Magazine accepted the Borland advertisement indicated a policy of accepting advertisements—even from competitors. Moreover, on the evidence before me I cannot find, as factfinder, that the plaintiff in this case, Lotus Development Corporation, controlled the decision-making process about what appears in Lotus Magazine. Accordingly, it was not reasonable for Borland to infer that Lotus Development Corporation was making any statement concerning possible copyright infringement when Lotus Magazine accepted the advertisement.

Borland next cites examples of occasions where Lotus personnel interacted with Borland personnel at trade shows and with respect to new products and services, without mentioning any belief that the Borland products infringe. Nothing in these materials suggests that the Lotus personnel involved had the actual or apparent authority, or even sufficient knowledge, to comment on whether or not Lotus believed that Borland

infringed a copyright or whether Lotus intended to file an action in the future. Moreover, nothing in the record shows that the individuals involved actually said anything that would support an inference by Borland that copying of the menu tree was permissible. Accordingly, I find that Borland could not reasonably interpret these communications to mean that Lotus would not file suit.

In sum, I find that both separately and taken as a whole, the events, conduct, and communications that Borland points to in an effort to bolster claims for estoppel fall far short of supporting estoppel.

C. Conclusion.

For the reasons explained above, I find that Lotus's claims in Phase I of the trial are not barred by laches or estoppel. In presenting its case on the affirmative defenses, Borland offers a number of proposed inferences that this court might draw, based loosely on evidence before the court. Although I have not explicitly referred to each of Borland's individual contentions in this Opinion, I have examined all of the contentions and have found them without merit.

UNITED STATES DISTRICT COURT
D. MASSACHUSETTS

Civ. A. No. 90-11662-K

July 31, 1992

LOTUS DEVELOPMENT CORPORATION,

Plaintiff,

—v.—

BORLAND INTERNATIONAL, INC.,

Defendant.

James C. Burling, Jeffrey B. Rudman, Hale & Dorr, Boston, Mass., Henry B. Gutman, Kerry L. Konrad, O'Sullivan, Graev & Karabell, New York City, *for plaintiff.*

Laura Steinberg, Sullivan & Worcester, Boston, Mass., Lynn H. Pasahow, McCutchen, Doyle, Brown & Enersen, San Francisco, Cal., David L. Hayes, Mitchell Zimmerman, Fenwick & West, Palo Alto, Cal., Peter Erich Gelhaar, Donnelly, Conroy & Gelhaar, Boston, Mass., Gary L. Reback, Wilson Sonsini, Goodrich & Rosati, Palo Alto, Cal., *for defendant.*

MEMORANDUM AND ORDER

KEETON, *District Judge.*

By Memorandum and Order of March 20, 1992, the court dismissed the parties' motions for summary judgment in this copyright infringement action and invited new motions compatible with rulings therein announced. Each party has renewed its motion for summary judgment and filed further

submissions (Docket Nos. 168-190).^{*} A hearing on these motions was held on May 19, 1992, and additional submissions were filed after that hearing.

The reader may find background information in two earlier documents issued by this court: the first, the Opinion in a related case involving the Lotus 1-2-3 program at issue here, *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F.Supp. 37 (D.Mass. 1990), and, the second, a Memorandum issued in this case, *Lotus Dev. Corp. v. Borland Int'l Inc.*, 788 F.Supp. 78 (D.Mass. 1992) (Memorandum and Order of March 20).

In the Memorandum and Order of March 20, I concluded that Lotus had failed to frame adequately its contentions with respect to the infringement of elements of its user interface less than the whole interface. In its renewed motion for summary judgment, Lotus asserts specifically that Borland has copied expressive elements of the 1-2-3 interface, including "menu commands," "menu structure," "long prompts," and "keystroke sequences." Although Lotus continues to argue that its entire user interface beyond dispute was copied, I adhere to the view that on the present record a reasonable jury could find that Borland copied less than the whole 1-2-3 user interface. Nevertheless, based upon the parties' most recent submissions, I conclude that, beyond *genuine* dispute, Fed.R.Civ.P. 56. Borland copied parts of the 1-2-3 user interface. For the reasons stated herein, I deny Borland's motion for summary judgment and grant, in part, Lotus' motion for summary judgment.

* Lotus argues that Borland has not filed a renewed motion for summary judgment. There is no docket notation of Borland's renewed motion, nor has such a motion been physically located by the court. However, several of Borland's submissions are submitted in support of "Borland's Renewed Motion for Summary Judgment." In this circumstance, and because I conclude that I must in any event deny such a motion, I will consider Borland's "motion" on its merits.

I. DEFINITIONS AND PREMISES

The Memorandum of March 20 presented for possible use in this case the following form of jury interrogatory concerning the extent to which Borland copied the Lotus 1-2-3 user interface in creating its Quattro programs:

Question 1

(a) Do you find that the Quattro Pro user interface as a whole was copied from the Lotus 1-2-3 user interface as a whole?

___ YES ___ NO

(b) Do you find that the part of the Quattro Pro user interface called the "emulation interface" (also called the "1-2-3 compatible interface") as a whole was copied from the Lotus 1-2-3 user interface as a whole?

___ YES ___ NO

(c) If NO, do you find that some part, and, if so, which of the following part or parts of the Lotus 1-2-3 user interface were copied into some part of the Quattro Pro "emulation interface" (also called the "1-2-3 compatible interface")?

- | | | |
|--------------------------|---------|--------|
| (1) The menu commands | ___ YES | ___ NO |
| (2) The menu structure | ___ YES | ___ NO |
| (3) The command sequence | ___ YES | ___ NO |
| (4) The long prompts | ___ YES | ___ NO |
| (5) The macro facility | ___ YES | ___ NO |

The Memorandum of March 20 noted that it was not clear that Lotus was making a claim of the sort addressed in part (a) of proposed Question 1 and that ambiguity remained regarding the meaning of the terms used in parts (c)(1)-(5). Lotus has in its recent submissions clarified its contentions.

First, Lotus acknowledges that the "native" modes of the Quattro programs have user interfaces that differ from that of 1-2-3. Thus, the question posed in part (a) of Question 1 is not in genuine dispute.

Second, Lotus has defined, as it uses them, the terms "menu commands," "menu structure," "keystrokes," "keystroke sequences," "long prompts," and "macro language."

In general, except for some blending of argument with definitions, the parties appear not to be in dispute about the meaning of these and other terms defined below. The definitions that I use in this Memorandum, for the purpose of explaining and analyzing the contentions of the parties, are consistent with the submissions of both Lotus and Borland, as I understand them.

"Command" refers to an abbreviated description of a direction that a user of a software program (whether Lotus 1-2-3, Borland's Quattro Pro, or another program) may invoke to cause some operation to be performed.

"Menu" refers to a display on the computer monitor of a limited number of commands available to the user at a given moment.

"Menu command" refers to a command that appears in a menu. In Lotus 1-2-3, a menu command is ordinarily a single English-language word. In rare instances, it is instead a representation of an English-language pronunciation (such as "Xtract"). Menu commands are displayed on the computer monitor by the 1-2-3 program in a succession of menus. The menus communicate to the user, in sequence, the spreadsheet operations available to the user.

"Command structure" refers to the organization of the menus and menu commands. (Other phrases used with essentially the same meaning include "menu command structure," "menu hierarchy," and "menu command hierarchy.") In Lotus 1-2-3, menu commands are organized so that less than a dozen related menu commands are displayed at any given moment. This display communicates to the user the spreadsheet operations immediately available. Each menu of less than a dozen commands is linked to preceding/succeeding menus by the operation of menu commands. All command menus are ultimately linked to a single main (root/trunk) menu to form a "menu tree."

"Keystroke sequence" refers to a sequence of keystroke entries that a user may invoke. Keystroke sequences may be

generated as one navigates the menu command hierarchy performing sequential spreadsheet operations.

"Long prompt" refers to a displayed multi-word English-language description of a "highlighted" menu command. A "highlighted" menu command appears on the computer monitor as a block of inverse video—that is, on a monochrome monitor with a black background on which characters are lit, a highlighted word appears as black letters within a lit block.

"Macro language" refers to a feature by which a user may define a very short keystroke sequence as equivalent to a longer keystroke sequence. Thus, a user may invoke the short keystroke sequence (a "macro") as a substitute for the longer keystroke sequence. In stating this definition, I omit a sophisticated programming capability available in 1-2-3 through its macro language feature that Lotus, as I understand its submissions, does not contend is involved in its claim of infringement in this action.

Having stated the definitions of the components of the user interface that I will use in this Memorandum, I now state additional points that I conclude are not in dispute about the relations among these definitions and associated matters.

The keystroke sequences and macro language have functionality. Typing ("inputting," in jargon) the first character of a command word invokes the command and causes the operation associated with the command word to be performed. (In many instances, a submenu associated with the command word is displayed.) The menu command hierarchy is a fundamental part of the functionality of keystroke sequences and the macro language. For example, the keystroke sequence "/RFC" directs the computer to format a range of numbers to appear as currency values because the character "/" initiates a command sequence, the character "R" implements the "Range" command, the character "F" implements the "Format" command, and the character "C" implements the "Currency" command.

It may be necessary to enter additional information to invoke a spreadsheet operation fully. For example, in order to implement an operation formatting a range of numbers as currency values, it is necessary to delimit the range of spread-

sheet cells to be formatted. In addition, it is necessary to specify the number of decimal places to be displayed. Variables such as range of cells and number of decimal places, for which values must be input each time an operation is to be performed, are called "parameters."

The authors of Lotus 1-2-3 made certain predictions about the value of each of the relevant parameters likely to be input for use with certain spreadsheet operations. Those predictions have been incorporated into Lotus 1-2-3 as suggestions; a user failing to specify a value for a parameter where it is necessary to supply one accepts the suggestion of 1-2-3's authors by default. The "/RFC" command set, for example, has associated with it two "default parameters." The default for range is "current cell," and the default for number of decimal places is two (i.e., dollars and cents). The user who prefers a different format (for example, whole dollars rather than dollars and cents) may enter a different number, zero rather than two (the default) to so indicate. In similar fashion, the user may supply a range different than the default value.

The menu structure will not permit the command "Currency" to be executed without first proceeding through the "Range" and "Format" commands. Indeed, inputting a "C" at a different point in the menu structure may cause a different command, such as "Copy," to execute.

The foregoing description identifies one way in which a menu command may be invoked—that is, by pressing the letter key corresponding to the English-language name of the command. A second way of invoking a menu command is to make use of the highlighting around a menu command. The user may use arrow keys on the keyboard to move the block of highlighting ("cursor") to an adjacent menu command. Depressing the "Enter" key (or "Return" or "<—" key, depending on the keyboard) invokes a highlighted command.

Because the macro language plays such a central role in the parties' contentions, and because it is an extraordinarily sophisticated element of Lotus 1-2-3, I recite some further examples of the use of the macro language.

A user may define a keystroke sequence with a macro by inputting the keystroke sequence in a spreadsheet cell and

assigning a macro label to that cell. For example, if a user enters the sequence "Hello" (the tilde, "~," stands in for the "Enter" or "Return" or "<—i" key) in spreadsheet cell A1 and assigns the macro label "\H" to cell A1, then the user may cause the word "Hello" to appear in any other cell by invoking the abbreviated keystroke sequence "\H" instead of the longer sequence "H," "e," "l," "l," "o," "Enter." (The reverse slash, "\," signifies the "Alt" key, pressed and held in place while another key is pressed. Thus, the keystroke sequence "\H" consists of pressing the "h" key while the "Alt" key is simultaneously depressed.)

The capability of the program to enter, for example, the word "Hello" in any spreadsheet cell when the "\H" keystroke sequence is entered is functional. It is not protectable by copyright, 17 U.S.C. § 102(b), and it is not the subject of this case.

At a more sophisticated level, a user may construct a macro that implements menu commands. For example, a user may input the keystroke sequence "/RFC" in cell A1 and attach a macro label such as "\C" to that cell. This defines "\C" in the macro language subject to later redefinition—as "/RFC". Invoking the "\C" keystroke sequence in any cell will cause the spreadsheet to format that cell to display numeric values as currency values. The ability of the computer to format the cell for currency values is functional. It is not copyrightable, *id.*, and it is not the subject of this case. The spreadsheet program is instructed to format the cell by the keystroke sequence "/", "R," "F," "C," "Enter," "Enter." That sequence invokes the menu commands "Range," "Format," and "Currency" (and accepts, by operation of the "Enter" key twice, two default parameters). Thus, the menu commands are an important part of the functionality of the macros. Lotus contends that the menu commands and the command structure are copyrightable expressive elements of the 1-2-3 user interface and that they are copyrightable expressive aspects of the macros. Borland contends that the macros, in their entirety, are an uncopyrightable system. These competing contentions, as well as others, are addressed below.

II. COPYING

In Borland's Mem. in Supp. of Renewed Mot. for Summ. J. (Docket No. 168), Borland states the following:

Furthermore, it is undisputed in the record that Borland did not copy the 1-2-3 menu command hierarchy directly from any 1-2-3 version, including Release 2.01. Rather Borland employees reviewed books about 1-2-3, Release 2.01, written by third-parties, which books contain schematic or menu-tree type representations of the 1-2-3 menu command hierarchy. Borland used these third-party menu trees to construct 123-compatible menu hierarchies in their own products.

But Borland employees did not copy those menu trees, even from the third-party books. Rather, they viewed the menu trees and implemented into their own products the relationship of functions depicted in those menu trees.

Id. at 15 (footnote omitted). Borland contends that, on these facts, indisputably Borland did not "copy." Borland's contention, however, is based on its idiosyncratic use of the word "copy," and is fundamentally wrong. Instead, Borland's admissions establish beyond dispute that Borland did copy the menu commands and command structure of Lotus 1-2-3.

Borland argues that the menu command hierarchy is a "set of functional relationships" that is nowhere displayed in the 1-2-3 user interface. Thus, Borland argues, it did not and could not have "copied" the menu command structure. That argument simply fails. It is an argument about a fact—copying or not. Its premise bears instead not on the fact-copying or not—but upon the legal issue of copyrightability. That is, the premise of Borland's argument is that the menu command structure must be fixed in a tangible medium to be copyrightable. *See* 17 U.S.C. §§ 101-102. The admitted fact that the Quattro programs duplicate the set of "functional relationships" of Lotus 1-2-3 and were designed to do so is conclusive against Borland on the issue of copying that set of functional relationships. Thus, Borland has admitted that it

intentionally incorporated into its user interface the 1-2-3 menu commands and menu command hierarchy.

Moreover, I reject Borland's tangible-medium argument as applied to this case. The argument would be relevant, if at all, to copyrightability rather than copying, but because Borland has made the argument as if it had a bearing on copying, I will digress briefly to address it here. To be the subject of copyright protection, an expression must be fixed in a tangible medium. *Id.* § 102. All that is required in this regard is that the expression be embodied in a copy "by or under the authority of the author" in a form "sufficiently stable to permit it to be perceived, reproduced, or otherwise communicated for a period of more than transitory duration." *Id.* § 101. The output of a computer program, at least insofar as it is typical of the program, predictable from it, and directed by the operation of the program, satisfies these requirements. The menu command hierarchy is part of the 1-2-3 program's output, is directed by the program, is identical each time the program runs, and may be perceived (and, as in this case, duplicated). It is irrelevant that the hierarchy cannot be perceived in its entirety at one moment (for example, in one screen display), just as it is irrelevant that the plot of a novel cannot be perceived from viewing one page. The menu command hierarchy is copyrightable subject matter.

Borland's argument that it copied from third-party sources rather than Lotus is equally without merit. Borland argues this point on the strength of the court's suggestion that Borland might prove that it did not copy Lotus' interface by demonstrating that instead it copied someone else's. If Borland had copied a third party's independently created menu command hierarchy that "happened" to duplicate Lotus' interface, Borland might be excused (at least from liability to Lotus, though not perhaps to the third party). Here, however, Borland has admitted copying from sources that, with or without permission from Lotus, copied from Lotus. Borland was aware that those sources copied from Lotus—Borland admits verifying the accuracy of the menu structure it generated from third-party sources by comparing the Quattro programs to 1-2-3. The fact that Borland used third-party sources as a means of

copying the Lotus 1-2-3 menu command hierarchy in no way excuses Borland's deliberate imitation of the Lotus menu structure.

Based on the foregoing, I conclude that, beyond genuine dispute, Borland copied the menu commands and menu command structure of Lotus 1-2-3. Moreover, Borland admits to copying the functionality of the keystroke sequences and macro language. By its own assertions, Borland's reason for copying the menu command structure was to obtain the benefit of its functionality. It follows that in fact Borland has admitted copying aspects of the keystroke command sequences and macro language that Lotus contends are expressive and copyrightable, although Borland chooses to describe what it did in another way.

Lotus has identified at this stage one additional element of its user interface that it claims Borland copied: the long prompts. It is clear from the record that the long prompts appearing in the Quattro programs differ in many instances from the long prompts in 1-2-3. On the other hand, in many other instances there is little or no difference. There is evidence that a Borland employee wrote the Quattro long prompts and did not "copy" from 1-2-3; however, she admits to looking at 1-2-3 to ensure that she correctly understood the Lotus commands. A reasonable jury could find on the basis of this evidence either way—that Borland did or that Borland did not copy the long prompts of 1-2-3. Therefore, I conclude that whether the long prompts were copied is a question for the jury. Moreover, because of this conclusion, it is apparent that a reasonable jury could find that Borland did not copy the 1-2-3 interface as a whole.

III. COPYRIGHTABILITY

Having concluded that Borland copied the menu commands and menu command hierarchy as well as the keystroke sequences and macro language, I now proceed to determine whether those aspects of the 1-2-3 user interface, taken together, are copyrightable.

A. Potential Fact Questions

I stated earlier my tentative conclusion that the application of the copyrightability standard is for the court and not a jury. *Borland*, 788 F.Supp. at 96. I invited counsel to respond to that tentative conclusion.

Lotus has fully endorsed the conclusion that copyrightability issues, at least in this case, are for the court. Borland, however, contends (in the alternative to Borland's own motion for summary judgment) that questions of copyrightability are for a jury. In response to the court's invitation, Borland proposes the following questions to be answered by a jury:

1. Does the Lotus 1-2-3 menu command hierarchy comprise a system, procedure or method of operation?
2. Is the 1-2-3 menu command hierarchy designed and used as a system for performing tasks using a spreadsheet program?
3. Are the command words of 1-2-3 and their order an inseparable part of a system for performing spreadsheet tasks?
4. Does the 1-2-3 menu command hierarchy enable a person to map out and execute a procedure for performing a particular spreadsheet task?
5. Is the 1-2-3 menu command hierarchy a means for issuing commands to the computer program to perform spreadsheet tasks?
6. Does the 1-2-3 menu command hierarchy provide a procedure or method of operating a spreadsheet program?
7. Was the 1-2-3 menu command hierarchy designed and arranged using functional rules or principles?
8. Was the 1-2-3 menu command hierarchy designed and arranged to maximize its efficiency and usefulness?
9. Is the 1-2-3 menu command hierarchy fundamental to a user's ability to execute macros written using 1-2-3?

10. Do the command words of 1-2-3 convey information to the user other than the choices of functions that are available?

11. Does the 1-2-3 menu command hierarchy explain to the user how to use the 1-2-3 program to perform spreadsheet tasks?

Liability Questions for the Trier of Fact (Docket No. 95) at 2-5. Under my rulings, it is irrelevant that the 1-2-3 interface includes functional elements or "comprises a system" so long as it also includes separable expressive elements. Thus, five of Borland's first six questions are irrelevant. Question 3, though relevant, is a mixed law-fact question that will not be asked of a jury for reasons previously stated. *Borland*, 788 F.Supp. at 94-96.

The seventh and eighth questions are more problematic. Those questions may have a bearing on whether the expressive elements of 1-2-3 are in fact separable from the functional aspects of the interface. As stated in the Memorandum of March 20, "[i]f the menu commands or menu command structure were dictated solely by functional concerns, then those elements may not be copyrightable." *Borland*, 788 F.Supp. at 97 (citing *Brandir Int'l, Inc. v. Cascade Pac. Lumber Co.*, 834 F.2d 1142, 1145 (2d Cir.1987)). Lotus argues that *Brandir* is not "strictly applicable" because computer programs are classified as "literary works" and not "pictorial, graphic, and sculptural works" by the Copyright Act. The law draws heavily on analogy, however, and computer programs, whatever their formal classification, like pictorial, graphic, and sculptural works, are useful articles. Moreover, though insisting that the burden on this question falls on Borland, Lotus acknowledges that any elements of its program that were functionally dictated are not copyrightable. Pl.'s Mem. of L. in Supp. of Renewed Mot. for Summ. J. on Infringement (Docket No. 171) at 34-35 n. 47.

Nevertheless, I conclude that no reasonable jury could find that the menu command hierarchy was limited to one or even several alternate designs at the time it was created. On the

basis of the evidence before me, a factfinder could conclude that some—but only some—subelements of the menu command hierarchy were functionally dictated. For these reasons, it is my tentative conclusion that there may be genuine fact disputes regarding the subject matter of Borland's seventh and eighth questions. Questions seven and eight, as formulated, however, are "evidentiary" rather than decisive or "ultimate" issues appropriate for use on a verdict form.

It remains to be determined whether an "ultimate question" on the subject matter of the seventh and eighth of Borland's suggestions can be formulated as an adjudicative fact question of the type that is normally the province of a jury and does not pose the substantial risks described in my earlier Memorandum, *see Borland*, 788 F.Supp. at 94-96.

Question nine assumes a premise that is fundamentally in error—an error that has been systemic in Borland's arguments throughout the course of this litigation. The problem, which may be described as a "chicken and egg" problem, is addressed in greater detail in Section C, below. At this point, I simply state my view that question nine should not be asked of a jury.

The answers to the last two questions are irrelevant. A "no" answer to question ten does not preclude a determination that the command words are expressive even though the expression may be limited—i.e., the words communicate the functions to which they are assigned. Similarly, the answer to question eleven is immaterial.

B. Legal Issues on Copyrightability

In the Memorandum and Order of March 20, 1992, I concluded that, "absent further guidance from higher authority before the date of trial," *Borland*, 788 F.Supp. at 89, I would apply the following standard for deciding copyrightability issues:

FIRST, in making the determination of "copyrightability," the decisionmaker must focus upon alternatives

that counsel may suggest, or the court may conceive, along the scale from the most generalized conception to the most particularized, and choose some formulation, some conception of the "idea," "system," "process," "procedure," or "method"—for the purpose of distinguishing between the idea, system, process, procedure, or method and its expression.

* * *

SECOND, the decisionmaker must focus upon whether an alleged expression of the idea, system, process, procedure, or method is limited to elements essential to expression of that idea, system, process, procedure, or method (or is one of only a few ways of expressing the idea, system, process, procedure, or method) or instead includes identifiable elements of expression not essential to every expression of that idea, system, process, procedure, or method.

THIRD, having identified elements of expression not essential to every expression of the idea, system, process, procedure, or method, the decisionmaker must focus on whether those expressive elements, taken together, are a substantial part of the allegedly copy-rightable "work."

Id. at 90 (quoting *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F.Supp. 37, 60-61 (D.Mass.1990)) (all emphasis omitted).

The Second Circuit has recently issued an opinion that bears significantly on issues of copyrightability and substantial similarity. In *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 1992 WL 139364, No. 91-7893, 1992 U.S.App. LEXIS 14305 (2d Cir. June 22 1992), the court announced an "Abstraction-Filtration-Comparison" test for determining "substantial similarity." The first two steps, "abstraction" and "filtration," are designed to define the idea of the program and to eliminate it (as well as other noncopyrightable subject matter) from further consideration. The test "serves 'the purpose of defining the scope of plaintiff's copyright.'" *Id.* (quoting *Brown Bag Software v. Symantec Corp.*, 960 F.2d

1465, 1476 (9th Cir.1992)). Thus, the first two steps of the Second Circuit's "substantial similarity" test concern what other courts and commentators have called "copyrightability." Only the third step, "comparison," addresses similarities between the copyrighted work and the allegedly infringing work.

To what extent does the Second Circuit's "Abstraction-Filtration-Comparison" test differ, either substantively or in methodology, from the combination of copyrightability and substantial similarity tests tentatively adopted for this case in my Memorandum and Order of March 20?

I conclude that the standard for determining copyrightability stated in my Memorandum and Order is compatible with the abstraction-filtration portion of the Second Circuit's test. The Second Circuit founded its abstraction step on the opinions of Judge Learned Hand that were also the foundation of the first step of the copyrightability test stated in my Memorandum and Order. The second step of that copyrightability test parallels the Second Circuit's "filtration" step.

The third step of the Second Circuit test, "comparison," serves two functions. The first concerns the issue addressed in the third step of the "copyrightability" test I have tentatively adopted for this case— whether the expressive elements of the allegedly copyrightable work are a substantial part of it. I conclude that in this respect the two tests are compatible substantively though different in methodology. The other function that the Second Circuit's "comparison" step serves is emphasized in the term used to identify it—"comparison." The comparison is between the relevant portions of the allegedly infringing work and the expressive elements of the allegedly copyrightable work to ascertain whether any part of the allegedly infringing work is similar to expressive elements of the allegedly copyrightable work that are a substantial part of the allegedly copyrightable work (i.e., whether there is substantial similarity in the mixed law-fact sense). I conclude, again, that in relation to this comparison, the Second Circuit's test and the combination of the "copyrightability" and "substantial similarity" tests I have adopted tentatively are compatible substantively, though different in methodology.

Borland argues that the decision and reasoning in *Computer Associates* are contrary to the *Paperback* decision, and, as well, to my Memorandum and Order of March 20. In one respect only, however, did the Second Circuit explicitly so indicate. It criticized an incentive-based reason stated at one point in the *Paperback* opinion. The criticized argument, however, was by no means essential to the outcome in *Paperback*, and acceptance or rejection of that argument is not likely to affect the outcome in this case. In other respects, my reasoning in *Paperback* and in my Memorandum and Order of March 20, 1992 in this case was substantively consistent with the opinion of *Computer Associates*, as I understand that opinion, though, as I have explained above, different in methodology.

A particular example of significance to this case concerns Borland's argument that the Second Circuit's treatment of "compatibility" militates against copyrightability of the 1-2-3 interface. Borland extracts from the Second Circuit's opinion a determination that aspects of computer software cannot be subject to copyright if they are greatly circumscribed by the hardware or software with which they are designed to interact. That proposition, even if accepted as Borland has stated it, does not apply to 1-2-3. Borland's argument to the contrary must be rejected for reasons that I explain in Part C, immediately below.

C. Which Came First?

A familiar childhood riddle asks: Which came first—the chicken or the egg? As folk riddles often do (and lawyer's questions on cross-examination sometimes do), this riddle strongly suggests that any answer but one of two identified options is out of bounds. Perhaps from another perspective, however, one may recognize that the chicken and egg are one type of organism in different stages of a life cycle. Over a long span of time the form of that life cycle has evolved from an earlier form in which the chicken and the egg were not so distinct as they now appear to be. History, science, and

philosophy may provide another answer, or many others, different from the two suggested in the riddle.

Borland's brief poses a conundrum for this case in a form analogous to the chicken-and-egg riddle. Borland asserts that the 1-2-3 interface is not copyrightable because the menu command hierarchy "was dictated by the nature of the user macros with which it was designed to interact." Supp.Mem.Re: Additional Authority (attached to Docket No. 189) at 9 (quoting *Computer Assocs. Int'l v. Altai, Inc.*, 1992 WL 139364, at *24, No. 91-7893, slip op. at 50, 1992 U.S.App. Lexis 14305, at *67 (2d Cir. June 22, 1992)). An implicit premise behind this argument is that the menu command hierarchy was designed to fit the macros. The subtle suggestion is that the macros came first—that they were pre-existing. Necessarily implicit in Borland's argument is the assertion that neither the final version of the menu command hierarchy nor any substantial part of it was preexisting when the macro language was created.

The thrust of the discussion of "compatibility" in *Computer Associates*, however, relies upon *proof* that what the program was designed to fit was already in existence before the program was designed to fit it. Thus, a program designed to fit hardware specifications cannot be protected by copyright unless the program contains expressive elements not substantially dictated by the hardware. Similarly, a program designed to interact with preexisting software, such as the operating system at issue in *Computer Associates* is not entitled to protection to the extent that it is constrained by the need for compatibility with the preexisting software. Thus, the rule makes sense if the premise of a preexisting functional limitation is valid.

In this case, however, there is a very simple answer to the question "Which came first?" The Lotus 1-2-3 interface—or at least a version of it—was written first. All user macros derive from it. Thus, Borland is simply wrong factually to argue that the 1-2-3 interface was constrained by the macros. On this issue, there is no genuine dispute of fact.

I assume in Borland's favor that, like the chicken and the egg, the macro language (as opposed to macros defined by

users using the macro language, which necessarily came later) evolved simultaneously with the menu commands that delimit it. Nevertheless, it is beyond dispute that the macro language did not evolve *first*.

Borland has argued, also, that the need to ensure that the menu commands in any given menu begin with a different letter is a functional constraint. Even that argument fails, however, because of the availability of other symbolic tokens; e.g., A, B, C, . . . or 1, 2, 3, . . .

It is no doubt true that the macros have functional significance. Moreover, as this court found in *Paperback*, the menu "system" is a fundamental part of the functionality of the macro language and the macros. As the Second Circuit has recognized, however, the fact that a form of expression takes on functional character does not remove it from the protection of copyright. *Brandir Int'l, Inc. v. Cascade Pac. Lumber Co.*, 834 F.2d 1142, 1147 (2d Cir.1987); *see also Paperback*, 740 F.Supp. at 58.

Lotus used the slash character ("/") to initiate a command sequence; the character "R," to specify the command "Range"; the character "F," to specify the command "Format"; and the character "C," to specify the command "Currency." A user inputting the characters "/", "R," "F," and "C" in sequence (and supplying necessary additional parameters) will cause the computer to perform the operations associated with the specified commands.

If the keystroke sequence "/RFC—" were stored in a spreadsheet cell that had been assigned the macro "\C," then a user inputting the "\C" keystroke sequence would cause the computer to perform the same operations as would be invoked by the six-keystroke sequence "/", "R," "F," "C," "Enter," "Enter." Because the "\C" keystroke sequence is invoked by pressing the "c" key with the "Alt" key depressed as one would use a "Shift" key, it involves two keys but perhaps only one "keystroke." Thus, depending upon one's definitions, a saving of either five or six keystrokes would result.

Had Lotus preferred, it could have chosen, for example, the command "Scope" instead of "Range," "Appearance" instead of "Format," and "Money" instead of "Currency." Then the

user would invoke the sequence “/,” “S,” “A,” “M” to establish a “money” appearance (\$xxx.xx) for a scope of cells. In that case, the parallel to Borland’s argument would be that the existence of “/SAM” macros in users’ files (spreadsheet files with cells defining macros may be saved for future use) “dictated” the use of the “Scope,” “Appearance,” and “Money” commands in the menu command hierarchy. Obviously, the argument is without merit—“Range,” “Format,” and “Currency” are demonstrably acceptable choices. The fact that Lotus chose one command set for its first version may have made it “necessary” that future versions adopt the same command set; however, the initial choice of the command set was a free choice.

It bears repeating here that Lotus (though not Borland) was entitled to incorporate in version 2.01 the menu command hierarchy that it employed in earlier versions of 1-2-3. Thus, Borland’s contention that the macros preceded and dictated the menu command hierarchy of version 2.01 fails because that fact is irrelevant. This is so because Lotus’ exercise of creative expression that is at issue in this case (the menu command structures of versions 1A and 2.0) manifestly preceded the users’ macros that incorporate the 1-2-3 command hierarchy.

It may be argued that the macros, by themselves, are functionally dictated by the 1-2-3 menu command hierarchy and so are not copyrightable independently of the copyrightability of the menu command hierarchy. Moreover, even if some macros are copyrightable, it may be that the owner of any copyright in a macro is the user who authored the macro and not Lotus. I am not faced with those questions in this case. I conclude only that Borland’s argument rests on a false proposition. Thus, to decide this case I need not and do not decide whether Borland is prohibited from reading and interpreting the macros that have been created by users of 1-2-3. Had Borland created a program that read users’ 1-2-3 macros and converted them to macros for use in the Quattro programs’ native modes, so that they could be interpreted, executed, modified, debugged, etc. by resort to Borland’s

command hierarchy, that would have presented a different case from the one now before me.

Borland did not obtain the right to expressive aspects of Lotus’ command hierarchy merely because—if it be the case—the 1-2-3 program revolutionized the spreadsheet market. The menu command hierarchy has a functional aspect when incorporated into the keystroke sequences and macros. That functional aspect is separable from the expressive aspect that preceded it. Borland cannot obtain the right to use the macros and keystroke sequences just because the only means of doing so is by infringing expressive features of the Lotus 1-2-3 macro language and keystroke sequences.

The error of Borland’s argument may be demonstrated by a simple hypothetical. If an uncopyrighted movie is made from a copyrighted novel (under the authority of an appropriate license), the public is free to copy all aspects of expression *unique to the movie*; however, the novel does not enter the public domain. Similarly, if the macros have uncopyrightable functional aspects, Borland does not infringe the copyright in Lotus 1-2-3 version 1A by duplicating the functionality in any way that does not copy the expressive elements of Lotus 1-2-3 version 1A, but it must not infringe upon expressive elements.

Borland’s argument that *Baker v. Selden*, 101 U.S. 99, 25 L.Ed. 841 (1879), establishes a different rule that controls this case is rejected. This is not a case like *Baker v. Selden* in which the system depends on the use of the copyrighted matter. Borland has, in fact, designed a system (Quattro Pro’s native mode), using macros and keystroke sequences and using an alternate command hierarchy, that is fully functional.

D. More on *Computer Associates v. Altai*

Some further points regarding the Second Circuit’s opinion bear upon its applicability to issues in the present case.

1. *Beyond Program Code to Nonliteral Expression.*

The reasoning underlying the Second Circuit's "Abstraction-Filtration-Comparison" test extends beyond program code to nonliteral expression. That is, although the issue in that case concerned program code and its structural aspects, the Second Circuit based its conclusions on reasons of broader application. In fact, the court explicitly approved determinations with respect to the copyrightability of certain nonliteral, noncode (nonstructural) aspects of the 1-2-3 spreadsheet in *Paperback*. See *Computer Assocs.*, 1992 WL 139364, at *16, No. 91-7893, slip op. at 34, 1992 U.S.App. LEXIS 14305, at *46. Thus, from statements made in the Second Circuit's opinion, albeit *obiter dicta*, I draw support for my conclusion that certain expressive elements of the 1-2-3 user interface may be protected by copyright.

2. *Criticism of Whelan.*

Borland's argument that the Second Circuit, with one stroke, knocked out the "conceptual underpinnings" of *Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222 (3d Cir.1986), *cert. denied*, 479 U.S. 1031, 107 S.Ct. 877, 93 L.Ed.2d 831 (1987) and of *Paperback* cannot be sustained. First, *Whelan* remains good authority in the Third Circuit and may provide guidance for this court in the same way as the Second Circuit's opinion in *Computer Associates*. That is, for courts in the First Circuit, each of these decisions from other federal circuits is instructive and neither is controlling beyond the persuasive force of its reasoning. Second, and more to the point, the Second Circuit criticized the *Whelan* decision for reasons that in large part do not apply to the rulings I made in *Paperback* or to the rulings I have made and now make in this case.

The Second Circuit was critical of *Whelan* on two principal grounds—that it failed to recognize that programs may have more than one idea, and that it mistakenly asserted that all that is not idea is expression. To resolve those problems, the Second Circuit formulated its "Abstraction-Filtration-Comparison" test.

In both these respects, the rulings I have made in the present case are compatible substantively (though not entirely in methodology) with the Second Circuit's views rather than those of the Third Circuit, where those two circuit's differ. For example, as indicated in the *Paperback* opinion, I recognize that the 1-2-3 user interface can be described as incorporating more than one idea. Thus, the *Paperback* opinion identifies the idea of an electronic spreadsheet, the idea of having a readily available method of invoking the menu at command system (which I concluded merged with its expression as the "/" command), and the idea of the two-line moving cursor menu, among others.

3. *Baker Not Alone Controlling.*

In its most recent submissions, Borland once again insists that 17 U.S.C. § 102(b) and *Baker v. Selden* control this case, now asserting that the Second Circuit decision supports Borland's argument. Once again, Borland overstates the point. Borland is correct in observing that the Second Circuit treats *Baker* as the "starting point" in analyses of utilitarian works, including computer programs. That view is compatible with my earlier Memorandum and Order. See *Borland*, 788 F.Supp. at 92-93. Borland fails to note, however, that the Second Circuit went on to state:

While *Baker v. Selden* provides a sound analytical foundation, it offers scant guidance on how to separate idea or process from expression, and moreover, on how to further distinguish protectable expression from that expression which "must necessarily be used as incident to" the work's underlying concept.

Computer Assocs., 1992 WL 139364, at *11, No. 91-7893, slip op. at 24, 1992 U.S.App. LEXIS 14305, at *32. The Second Circuit's decision cannot fairly be characterized as holding that *Baker* controlled the outcome in *Computer Associates*. The Second Circuit was sensitive not only to its duty of fidelity to precedent, but as well to its duty of fidelity to congressional mandates that came into existence long after *Baker* was decided.

4. "Substantial Similarity" and "Copyrightability."

The Second Circuit referred to its test as one of "substantial similarity" and did not use the term "copyrightability" for any part of the test. I do not understand this difference of terminology to have substantive implications, however, and more especially, not for this case. Nor do I understand the Second Circuit's three-step test to be meant as a rigid barrier to alternate methods of analysis and decision. The court "advised" rather than mandated that the courts within the Second Circuit adhere to the three-step test announced. Moreover, a court in the First Circuit must take account of some degree of dissonance between the First and Second Circuits regarding the methodology of infringement analyses. Compare *Computer Assocs., supra, with Concrete Mach. Co. v. Classic Lawn Ornaments, Inc.*, 843 F.2d 600 (1st Cir. 1988). Although *Concrete Machinery* outlines a methodology different from the three steps of the Second Circuit's test, I do not understand the First Circuit to have mandated the order of analysis it described. See *Borland*, 788 F.Supp. at 86. I adhere to the view stated in my Memorandum and Order of March 20 that, under existing precedents, prudential concerns about case management may have a bearing on the order in which a court proceeds with its analysis in a complex copyright infringement case. *Id.* Thus, I conclude that I may and should proceed to a copyrightability analysis at this time even though some potentially material factual questions are still unresolved—facts regarding the long prompts, and facts regarding the extent to which the menu commands and command hierarchy may have been dictated by functional considerations.

E. Applying the Standard

1. The First Step.

FIRST, in making the determination of "copyrightability," the decisionmaker must focus upon alternatives that counsel may suggest, or the court may conceive, along the scale from the most generalized conception to

the most particularized, and choose some formulation, some conception of the "idea," "system," "process," "procedure," or "method"—for the purpose of distinguishing between the idea, system, process, procedure, or method and its expression.

One may describe a number of conceptions of the 1-2-3 user interface. A non-exclusive list, commencing with the most abstract and moving toward the particular, includes:

- (1) Lotus 1-2-3 is an electronic spreadsheet.
- (2) It is a menu-driven electronic spreadsheet.
- (3) Its user interface involves a system of menus, each menu consisting of less than a dozen commands, arranged hierarchically, forming a tree in which the main menu is the root/trunk of the tree and submenus branch off from higher menus, each submenu being linked to a higher menu by operation of a command.
- (4) Its user interface involves a system of menus, each menu consisting of less than a dozen commands, arranged hierarchically, forming a tree in which the main menu is the root/trunk of the tree and submenus branch off from higher menus, each submenu being linked to a higher menu by operation of a command, so that all the specific spreadsheet operations available in Lotus 1-2-3 are accessible through the paths of the menu command hierarchy.
- (5) Finally, one may conceive of the interface as that precise set of menu commands selected by Lotus, arranged hierarchically precisely as they appear in 1-2-3. Under this conception, the interface comprises the menu of commands "Worksheet," "Range," "Copy," "Move," "File," "Print," "Graph," "Data," "System," and "Quit," linked by operation of the command "Worksheet" to the menu of commands "Global," "Insert," "Delete," "Column," "Erase," "Titles," "Windows," "Status," and "Page," etc. (The completion of this proposed statement of the "idea," listing all of the more than 400 commands for which "etc." stands, would require several dozen more lines of text.)

Borland argues that the appropriate conception of the "idea" of the 1-2-3 interface is the fifth option. If that were the case, of course, there would be no elements of expression in the menu commands and menu command hierarchy and therefore no copyrightable aspects in them. The premise of Borland's argument is that an "idea" of Lotus 1-2-3 version 2.01 is complete compatibility with earlier versions of 1-2-3, and more precisely with macros generated for use with earlier versions. Borland argues that the precise menu commands and menu structure are necessary to such functional compatibility. Thus, the argument goes, the entire interface of version 2.01 is a functional system or "idea" and is not copyrightable. This argument is essentially tautological. As applied to any case involving a useful article, an argument of this kind would always define the idea to incorporate all the specifics of the particular expression of that idea in the allegedly copyrightable work. Nothing would be copyrightable under this methodology of analysis. The argument is an attempt to win by definition without focusing at any time on any substantive issue concerning the separation of idea and expression.

To select, at the opposite extreme, the very abstract statement of the idea of 1-2-3 as "an electronic spreadsheet" would be to draw an inappropriately abstract boundary between idea and expression. Thus, I concur in a fundamental principle of the *Computer Associates* opinion and reject the contrary proposition in *Whelan*.

Arguably, my Opinion in the *Paperback* decision, where no sharper focus was essential to the outcome, is consistent with accepting a conception of the idea that falls between the second and third formulations above. See *Paperback*, 740 F.Supp. at 67 (electronic spreadsheet having "menu structure"). In any event, I now explicitly recognize that for decision of the issues now before me the selection of functional operations that the spreadsheet performs must be considered part of the idea of the program. Copyrightability depends on expression distinct from the selection of the set of spreadsheet operations that can be performed.

I conclude that an appropriate conception of the "idea" or "system" of the 1-2-3 interface is the fourth of the five alternative conceptions stated above.

2. *The Second Step.*

SECOND, the decisionmaker must focus upon whether an alleged expression of the idea, system, process, procedure, or method is limited to elements essential to expression of that idea, system, process, procedure, or method (or is one of only a few ways of expressing the idea, system, process, procedure, or method) or instead includes identifiable elements of expression not essential to every expression of that idea, system, process, procedure, or method.

Does the Lotus 1-2-3 user interface include identifiable elements of expression? For reasons stated below, I conclude that it does.

A very satisfactory spreadsheet menu tree can be constructed using different commands and a different command structure from those of Lotus 1-2-3. In fact, Borland has constructed just such an alternate tree for use in Quattro Pro's native mode. Even if one holds the arrangement of menu commands constant, it is possible to generate literally millions of satisfactory menu trees by varying the menu commands employed.

This may be easily demonstrated. Recall the ten commands that appear in Lotus' main menu: "Worksheet," "Range," "Copy," "Move," "File," "Print," "Graph," "Data," "System," and "Quit." One can imagine an entirely plausible spreadsheet in which the "Worksheet" command has been named, quite naturally, "Spreadsheet." Of course, this might require changing the "System" command to avoid two commands abbreviated "S," perhaps to "DOS." The "Quit" command could be named "Exit" without any other modifications. The "Copy" command could be called "Clone," "Ditto," "Duplicate," "Imitate," "Mimic," "Replicate," and "Reproduce," among others (in some cases requiring modification of other com-

mands in the menu). Additional possibilities include "Output" for "Print," "Draw" or "Chart" for "Graph," "Figures" or "Information" for "Data," "Scope" for "Range," and "Transfer" or "Relocate" for "Move."

Just these potential modifications of the main menu yield over 250 combinations of commands in the main menu with ten distinct first letters. Changes in submenus increase the number of possible menu hierarchies *geometrically*. Since there are dozens of independent submenus, the number of possible menu hierarchies is extremely large.

Borland argues that "[t]o hold that an idea, plan, method or art described in a copyright is open to the public but that it can be used only by the employment of different words and phrases which mean the same thing, borders on the preposterous." Borland's Resp. to Pl.'s Renewed Mot. for Summ. J. (Docket No. 183) at 19 (quoting *Crume v. Pacific Mut. Life Ins. Co.*, 140 F.2d 182, 184-85 (7th Cir.), *cert. denied*, 322 U.S. 755, 64 S.Ct. 1265, 88 L.Ed. 1584 (1944)). This case, however, unlike *Crume*, is not a case in which the system "can be effected solely by the employment of words descriptive thereof." *Crume*, 140 F.2d at 184. Use of just the initial letters of command words (together with long prompts) or of other symbolic tokens would have been a sufficient alternate method of implementing the system. In this case, the command words chosen are not necessary to expression of the system nor are they necessarily incident thereto. See *Computer Assocs.*, 1992 WL 139364, at *13, No. 91-7893, slip op. at 30, 1992 U.S.App. LEXIS 14305, at *40.

Lotus argues that a large number of substantially different arrangements (hierarchies) could also have been effected. Looking again at just the main menu, is there any reason that the commands "Copy" and "Move," for example, could not have been arranged in the opposite order? Borland argues that the arrangement was necessary, citing evidence that Lotus arranged the menu commands in order of the expected frequency of use.

It is clear that certain command words have been grouped according to function; e.g., there are eighteen commands that

affect the display mode of spreadsheet cells that are grouped together under the "Format" command. Thus, a jury could find that at least some aspects of the arrangement of command words, as opposed to the specific choice of command words, was guided by functional concerns.

This is, however, a disputable fact question that may affect only the scope of relief in this case. See *ABKCO Music, Inc. v. Harrisongs Music Ltd.*, 508 F Supp. 798 (S.D.N.Y.1981) (court awarded damages based upon contribution to success of infringing work of copyrighted material) (for subsequent history, see 944 F.2d 971 (2d Cir.1991)). I conclude that it cannot be genuinely disputed that a large part of the structure and arrangement of the menu commands is not driven entirely by functional considerations. There are sufficient non-functional aspects that at least hundreds and perhaps thousands of different expressions of the function were possible when Lotus chose the particular structure of menu commands incorporated into Lotus 1-2-3.

This may be demonstrated by examining more closely Borland's argument that the menus were arranged according to the predicted frequency of use of the commands. I assume the truth of Borland's assertion that Lotus predicted before marketing its spreadsheet that the "Copy" command would be used more often than the "Move" command. Nevertheless, that is merely a prediction of frequency of use. It did not require Lotus to list "Copy" before "Move." A user can type a "C" or an "M" with equal ease no matter which command is listed first. If a user prefers to invoke a command by first highlighting it and then typing "Enter," "Move" is only one keystroke from "Copy"; moreover, the same "right arrow" key that takes the cursor from "Worksheet" to "Copy" moves the cursor from "Copy" to "Move." Thus, the order in which commands are listed in a menu has very limited functional value.

In addition, a prediction of frequency of use depends upon who is the predicted user and what the predicted uses are. For example, a user may work on a spreadsheet without printing any of the work performed on that day. Yet, the user will ordi-

narily invoke the "Quit" command appearing at the end of the main menu at the end of each session. Nevertheless, the "Print" command appears before the "Quit" command on the menu.

Many users may rarely invoke the "System" command available in 1-2-3. That is, the need to invoke a disk operating system ("DOS") "shell" from within the spreadsheet program may be for most users of 1-2-3 a rare event. If the commands in the main menu of 1-2-3 are listed in order of predicted frequency of use, why does the "System" command "precede" the "Quit" command? In fact, what does it mean to say "precede"? "Q" neither precedes nor follows "S" on the keyboard. "System" precedes "Quit" as one moves from left to right within the main menu using the "right arrow" key. It is also true that "Quit" precedes "System" as one moves from right to left using the "left arrow" key. The 1-2-3 menus are circular (in jargon, they have a "wraparound" feature)—moving the cursor one step beyond the "end" of the menu causes the cursor to come to rest at the opposite "end." Thus, from the default cursor position on the "Worksheet" command, arguably "Quit" precedes "System."

For all these reasons, any *ex ante* prediction of frequency of use is itself of limited usefulness. It follows that the arrangement of menu commands according to predicted frequency of use is not a major functional limitation on the number of arrangements of menu commands.

The menu command hierarchy is an integral part of the functionality of the macros and of the keystroke sequences. Nevertheless, the fact that the macros and keystroke sequences incorporate the menu command hierarchy into their functionality does not remove the menu command hierarchy from the scope of copyright, if otherwise subject to copyright protection. Moreover, the macros and keystroke sequences are protected to the extent that it is necessary to infringe a copyright to use them. Of course, as I have stated above, it was *not* necessary to copy expressive aspects of the macro language and keystroke sequences to copy their function.

3. *The Third Step.*

THIRD, having identified elements of expression not essential to every expression of the idea, system, process, procedure, or method, the decisionmaker must focus on whether those expressive elements, taken together, are a substantial part of the allegedly copyrightable "work."

The question posed by this element of the copyrightability test is whether the creativity involved in establishing the menu commands, menu command hierarchy, macro language, and keystroke sequences was more than trivial. No reasonable jury could find otherwise. As Borland has itself acknowledged, at least implicitly, Lotus 1-2-3 was a dramatic change and improvement over what was available on the market at the time Lotus was created. Although a large portion of that improvement relates to the functional aspects of Lotus 1-2-3, the features that I have now concluded are expressive also played a substantial role. Borland has maintained that those features are part of an uncopyrightable system (an argument I now reject for the reasons stated), but Borland has never argued that they were trivial, nor could it do so persuasively.

F. *Ashton-Tate v. Ross*

Borland argues that the Ninth Circuit's decision in *Ashton-Tate Corp. v. Ross*, 916 F.2d 516 (9th Cir.1990), *aff'g* 728 F.Supp. 597 (N.D.Cal.1989) militates in favor of a conclusion that the menu commands and command hierarchy are not copyrightable. The Ninth Circuit held that a list of menu commands was not copyrightable for reasons stated by the district court. The district court, in turn, held that a document bearing "a list of labels for user commands, many of which are common commands that were already available on other software programs" was not innovative or novel. *Ashton-Tate Corp. v. Ross*, 728 F.Supp. 597, 602 (N.D.Cal.1989). The relevance of that conclusion to this case is in some doubt in view of the fact that Lotus 1-2-3 is one of the programs on which