No. 94-2003

IN THE SUPREME COURT OF THE UNITED STATES

October Term, 1995

Lotus Development Corporation,
Petitioner,
v.
Borland International, Inc.,
Respondent.

On Writ of Certiorari to
the United States Court of Appeals
for the First Circuit

MOTION FOR LEAVE TO FILE BRIEF AMICUS CURIAE AND BRIEF AMICUS CURIAE OF COMPUTER SCIENTISTS IN SUPPORT OF RESPONDENT

Ron Kilgard
(Counsel of Record)
Karl M. Tilleman
Dalton, Gotto, Samson & Kilgard, P.L.C.
Suite 900
3101 North Central
Phoenix, AZ 85012
Phone: (602) 248-0088

Attorneys for Amici

December 1995

No. 94-2003

IN THE SUPREME COURT OF THE UNITED STATES

October Term, 1995

Lotus Development Corporation,
Petitioner,
v.
Borland International, Inc.,
Respondent.

On Writ of Certiorari to
the United States Court of Appeals
for the First Circuit

MOTION FOR LEAVE TO FILE BRIEF AMICUS CURIAE
OF COMPUTER SCIENTISTS

The computer scientists identified in Appendix A to the attached brief respectfully move, pursuant to Supreme Court Rule 37.3(b), for leave to file the attached amicus curiae brief in support of Respondent Borland International, Inc. ("Borland"). The consent of the attorneys for Borland has been obtained. The consent of the attorneys for petitioner was requested but refused. In support of our motion, we respectfully represent the following:

1. As computer scientists who have invented or contributed to the authorship of numerous computer programs and languages, as well as many computer-related works, we appreciate the value of copyright protection as a means of encouraging as well as stifling innovation.

2. We have become increasingly concerned about the controversy and uncertainty in the computer industry regarding the legal rules for software development, and wish to communicate our views on the important issues in this case, which may have a significant impact on the computer software industry.

3. As stated in the accompanying brief, copyright

protection for computer interfaces and languages not only is legally unfounded, but also unwise for a variety of policy and industry-related reasons. Human/computer interfaces are distinct from the text of computer programs; they are the boundaries shared by the computer system and the human user. Communication across these boundaries is necessary for operation of a computer system. Computer languages are as distinct from the text of computer programs as the English language is from the text of a novel or a play.

4. We believe that granting protection to computer interfaces and languages will discourage innovation in the computer industry, which depends upon the ability of people and computers to communicate freely with each other, using many different computers and computer programs. As a result, software compatibility and interoperability are extremely important. Once a computer interface is developed, it has the potential to become an industry standard, becoming functionally embedded not just in the original work, but also in subsequent works. Allowing the creator or the "first comer" to use the copyright laws to restrict use of the standard would give the creator the ability to lock up the technology and the marketplace for 75 years, even if the interface and technology were unpatentable, resulting in enormous barriers to entry and discouragement of the development of creative works in the industry.

5. The copyrightability of computer interfaces is of paramount importance to the computer industry. As a group of scientists who have invented or contributed to the authorship of numerous computer programs and languages and who will be profoundly affected by the decision of the Court, we believe we are in a unique position to address the dangers of granting copyright protection to computer interfaces more completely than it will be addressed by the parties.

For the foregoing reasons, we respectfully request the Court grant our motion for leave to file the accompanying amicus curiae brief in support of Borland.

Respectfully submitted,

Ron Kilgard
(Counsel of Record)
Karl M. Tilleman

Dalton, Gotto, Samson & Kilgard, P.L.C.
Suite 900
3101 North Central
Phoenix, AZ 85012
Phone: (602) 248-0088

Attorneys for Amici
Listed in Appendix A

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Page(s)

CASES

STATUTES

MISCELLANEOUS

No. 94-2003

IN THE SUPREME COURT OF THE UNITED STATES

October Term, 1995

Lotus Development Corporation,
Petitioner,
v.
Borland International, Inc.,
Respondent.

On Writ of Certiorari to

the United States Court of Appeals
for the First Circuit

BRIEF AMICUS CURIAE OF
COMPUTER SCIENTISTS
IN SUPPORT OF RESPONDENT

I. INTRODUCTION AND INTEREST OF AMICI

This amicus brief is filed on behalf of numerous individual computer scientists who believe the opinion of the U.S. Court of Appeals for the First Circuit was correctly decided and should be upheld. The signatories to this brief include some of the leading, pioneering scientists in the computer industry. We have invented or contributed to the authorship of numerous computer programs and languages.n1 We have prepared this brief because we believe the First Circuit wisely corrected the District Court's error in overextending copyright coverage in a manner irreconcilable with the purpose of Copyright law and the nature of computer science.

Many of us filed an amicus brief in the First Circuit Court of Appeals, in which we discussed the copyrightability of computer languages, and urged reversal of the District Court's rulings. The District Court's opinions had generated widespread controversy and uncertainty in the computer industry because they went far beyond earlier case law, which recognized copyright coverage for the text of a computer program ("code") or for the visual, pictorial displays of the computer screen, but not for "computer processes." The District Court, however, held that functional aspects of Lotus' 1-2-3 spreadsheet program, and the program's procedures and methods of operation, were copyrightable and infringed, even though Borland copied neither the spreadsheet's programming code nor its visual displays. Specifically, the District Court held as copyrightable the program's menu commands, which are used to operate the computer, and which are used as a "macro language" for programming computer routines to execute a desired computer operation repeatedly. In protecting these computer processes, the District Court ignored the distinction between an expressive work (such as a novel or a computer program's "code"), on the one hand, and the language or medium used to express a work (such as the English language or a computer programming language), on the other hand. The First Circuit remedied that legal error by holding the menu command hierarchy to be an uncopyrightable method of operating the computer.

Not only does the First Circuit's opinion properly interpret the copyright laws, it also makes sense from a practical point of view. Fundamentally, this case directly affects the way people operate machines such as computers, and the way people communicate with computers using programming languages. This amicus brief explains two ways in which a user or programmer would use the menu command words of Lotus 1-2-3, namely (1) as part of the human/computer interface to run the program, just as buttons or switches were historically used to run machines, and (2) as a computer language to write macro programs. Such standardized computer interfaces and computer languages are highly important. Progress in the computer industry depends upon software compatibility and the ability of computer programmers to communicate freely with each other, using common computer interfaces and computer languages. Affirmance of the First Circuit's decision will help further these important policy goals. Because the copyrightability of computer interfaces and languages is a crucial issue to the computer industry, we respectfully submit the following brief on why computer interfaces, such as the Lotus 1-2-3 menu command hierarchy and macro language, should not be copyrightable.

II. SUMMARY OF THE ARGUMENT

1. Profound historical changes have occurred over the last several decades in the way in which people communicate with machines. Buttons, switches, and electrical plugs have been replaced with computers, and a variety of human/computer interfaces. One such interface consists of the Lotus 1-2-3 menu command words and the sequencing or syntax of those words, or "menu command hierarchy." Lotus uses menu words such as "COPY" or "PRINT" in place of buttons or switches which accomplished analogous functions on earlier machines.

2. Buttons, switches, and plugs have never been considered copyrightable on their own. Replacing them with their computer equivalents should make no difference under the law. This conclusion holds true even though a "computer program" is involved. Computer interfaces or languages are not the same as the computer program itself, which is the only aspect of a computer Congress has said should be copyrightable. Instead, a computer interface is an uncopyrightable method of operating a computer, as the First Circuit correctly found here.

3. The First Circuit's opinion also recognizes the reality faced by the computer industry. Standardization of computer interfaces and languages is vitally important to this industry. Requiring each new programmer to write a new interface or language would be absurdly inefficient and would decrease, not increase, the number of creative works in the industry.

4. In this case, the Lotus 1-2-3 menu command words and the sequencing or syntax of those words also define a computer language in which users can write computer programs known as "macros." As a computer language, the Lotus menu words and order are not the same thing as a computer program. It is very important to distinguish a copyrightable computer program from the uncopyrightable language in which the program is written. Languages are the building blocks used to create works such as novels or computer programs; protecting those building blocks is quite different from protecting the expression in a work created with those building blocks. Protecting a novel written in the English language is not the same as protecting the English language itself. Before this case, no one would have seriously argued that any language could be copyrightable, let alone a computer language. There are a number of important policy reasons, discussed below, why computer languages should remain uncopyrightable.

III. ARGUMENT

A. General Discussion of Computers

The following is a basic discussion of computers helpful to understanding the issues discussed in this brief. Broadly speaking,
computers have two components: hardware and software.n2 Hardware consists of the physical components used to make a computer, and includes three basic elements: the central processing unit ("CPU"), memory, and the input/output system. The CPU is the "brains" of the computer where the actual computing is done. On modern personal computers, the CPU is contained within a single silicon "chip" called a microprocessor. The CPU accesses various portions of the computer"s memory to store and help run computer programs.

Software generally refers to computer programs, which are defined in 17 U.S.C. sect. 101 as "a set of statements or

instructions to be used directly or indirectly in a computer to bring about a certain result." Broadly speaking, there are two types of software: operating systems software and application programs. The operating system software is the basic set of instructions that allows the CPU to function as a computer, manages the interaction among the various hardware elements of the computer and between the hardware and the application programs, and allocates the computer"s resources among other programs that might need them. The computer cannot function without operating system software. Because the operating system software must interact with other programs and with hardware components, the other programs must be compatible with the computer"s operating system software, that is, they must be able to exchange information with the operating system in a precise and accurate manner. See, e.g., Computer Associates, 775 F.Supp. at 550. Examples of operating systems are DOS, Windows 95, and UNIX.

Once a computer has an operating system installed, it can accept additional computer programs, called applications programs, which perform applications such as word processing, payroll and inventory management systems, graphics, video games, or, as in this case, computer spreadsheets.

When human beings write computer programs, they generally do so in human readable code, known as source code. Source code is written in computer languages such as assembly language, BASIC, or C++. In order for the computer to understand a program, a source code program must be translated into the "1s" and "0s" which are understandable to computers. This is called object code, and is virtually impossible for most humans to read.

B. Human/Computer Interfaces Should Not Be Copyrightable

1. Overview of Human/Computer Interfaces

This brief now explains the two ways in which a person running the Lotus 1-2-3 spreadsheet uses the Lotus menu command words, and their order. First, the person running the Lotus program uses the menu words to operate the spreadsheet. Therefore, the words and syntax are part of what is called the human/computer interface, or user interface. Generally speaking, an interface is a shared boundary, or the point at which a connection is made between two different elements or functional units in a computer system so that they can work with one another.

See Microsoft Dictionary at 218; IBM Dictionary at 351. The user interface consists of the graphical design, the commands, prompts, and other devices that enable a user to interact with and perform operations on a computer system or program. See Microsoft Dictionary at 218; IBM Dictionary at 724. The Lotus menu commands meet these definitions, as they are directions to the computer to "manipulate and control the program." Lotus Development Corp. v. Borland International, Inc., 49 F.3d 807, 809 (1st Cir. 1995), Pet. App. at 4a ("Lotus v. Borland"). More specifically, the Lotus menu command words and syntax are the "specification" for the human/computer interface. This means that the human operator must use the exact words in their exact order to operate the Lotus spreadsheet. No other words or order will do.

On a steadily increasing basis, human interaction with machines is now accomplished by human/computer interfaces. The First Circuit"s analogy to the video cassette recorder (VCR) is one example. Lotus v. Borland, 49 F.3d at 817, Pet. App. at 18a. Historically, users pressed buttons to operate electronic devices such as VCRs. The first VCRs were not computer controlled; pressing the buttons operated electronic circuitry which in turn performed record, play, and other functions. Timers and clocks were largely mechanical. Modern VCRs have replaced those electrical and mechanical components with a computer. The user can still operate many of the VCR functions by pressing buttons that still say "record" or "play." Thus, the computer/human interface looks the same as it did 20 years ago; the only difference is that the buttons now operate a computer instead of electrical or mechanical components.

Similarly, the manual typewriter has been replaced by a computer keyboard. It still uses the same QWERTY computer/human interface, even though that interface was designed to be used with the relatively inefficient manual typewriter and is far from the optimum design for a computer keyboard. Paul A. David, "CLIO and the Economics of QWERTY," 75 Am. Econ. Rev. 332 (1985). Nor are contemporary computer keyboards just a collection of hardware buttons. Modern word processing programs typically have features permitting the keyboards to be redefined, in software, so that the keys can represent different letters or functions than they normally do. See, e.g., User"s Guide for WordPerfect for Windows 6.1 at 305-308 (Novell, Inc. 1994) (discussing "Keyboard Preferences" and "Keyboard Editor" features).

Computers also have progressed to the point where software "buttons" have replaced hardware buttons or switches. Lotus 1-2-3 is such an example; commands such as "COPY" or "PRINT" appear on the screen, instead of a physical button. Lotus v. Borland, 49 F.3d at 817, Pet. App. at 18a-19a. Other programs use "touch screen" menus where the user can touch a word displayed on a computer screen and operate the desired function.

Moreover, technology has progressed to the point where no manual interaction between a human being and a computer is even necessary to operate the interface. At least three companies, including Lotus" owner, IBM Corp., are selling voice dictation and voice command systems enabling the user to speak the commands to the computer instead of either typing them, using function keys, or using a mouse.n3

2. Reasons Why Copyright Protection Is Not Appropriate

Twenty years ago, a company such as Lotus could not stop a competitor from using the organization of the buttons and switches on its product merely by paying $20 and filing a two-page form with the Library of Congress. Today, it is very difficult to see why Lotus should be allowed to use the copyright laws to stop its competitors from using the same command words of the Lotus program in the same order, when those words perform precisely the same function as a set of buttons did twenty years ago. The First Circuit noted that hardware buttons would not be copyrightable simply because of the "useful article" exception in the definitions of the Copyright Act, 17 U.S.C. sect. 101. Lotus v. Borland, 49 F.3d at 817, Pet. App. at 19a. As scientists, it is extremely difficult to understand why replacing hardware buttons with software buttons logically should change a formerly uncopyrightable "useful article" into something that is copyrightable, just because a "computer program" is involved. A human/computer interface, or a computer language, cannot be viewed in the same manner as any ordinary "literary work" exempt from the useful article exception of sect. 101. Someone can read an ordinary literary work, such as a novel or a play, and enjoy it for what it is worth. Someone can also read the text of a computer program written in source code as a work of authorship and gain some technical knowledge in the same way as one might gain such knowledge by reading a book about computers.

Computer interfaces are fundamentally different from any

of these. They are singularly functional and utilitarian in nature. One does not normally "read" a set of commands such as "COPY" or "PRINT," or the syntax or structure of those commands, as one might read a book about computers. Rather, the purpose of human/computer interfaces is to operate functions of the computer machine itself, and to use the computer.

Because of the nature of human/computer interfaces, there are a number of policy reasons why they should not be the subject of copyright protection. First, factors such as software compatibility, interoperability, and hardware portability are extremely important in the computer industry. Computers essentially have replaced the methods we used to operate machines in the past (hardware buttons or switches) with identically functioning buttons that instead operate software. In some instances, the physical switches appear the same to the user; in other instances, they are words on a computer screen.

This trend will only accelerate in the future. With computing hardware becoming increasingly inexpensive, software "virtual machines" have replaced electronic hardware in entire sectors of the economy. (A software "virtual" machine is one where the functions of electronic or mechanical components are primarily performed by software.) The ramifications of this computer revolution are enormous. The number of different kinds of computer/human interfaces have dramatically increased, both in sheer quantity as well as in the many different ways in which humans now interact with computers on a daily basis. Voice commands, and doubtless other non-physical methods of communication not presently imaginable, will increasingly replace hardware and even software buttons and switches.

The ability of programmers to write software programs that can be understood by other computers worldwide, on many types of computer platforms, is crucially important. The key point is that once an interface is created, it can become an industry standard, and the interface becomes functionally embedded not just in the original work, but in programs, and in the minds of programmers. The QWERTY typewriter keyboard and the positioning of automobile controls such as the gas and brake pedals and gear shift patterns are hardware examples of such standards; common computer languages such as BASIC and C++ are examples in the software industry.

Once a standard interface has been created, it would be absurdly inefficient to require all subsequent authors of computer

software to create their own interface with each new program. Imagine if each author of a novel were required to invent a new language because a prior author had a copyright in the English language. This would lead to less works of authorship being created, rather than encouraging the proliferation of works as the copyright laws were intended to do. See Fogerty v. Fantasy, Inc., 510 U.S. __, 127 L.Ed.2d 455, 465, 114 S.Ct. 1023, 1029 (1994) (noting the ultimate aim of the Copyright Act is "to simulate artistic creativity for the general public good").

Thus, granting copyright protection in computer interfaces would allow the creator or "first comer" to create a computer interface and use it to lock up the technology and marketplace for 75 years, even if the interface and technology was unpatentable. This would create enormous barriers to entry in a market where the free exchange of ideas is essential. See, e.g. Computer Associates International, Inc. v. Altai, Inc., 982 F.2d 693, 712 (2d Cir. 1992), citing P. Menell, An Analysis of the Scope of Copyright Protection for Application Programs, 41 Stanford L. Rev. 1045, 1082 (1989). This effective monopoly on a computer interface would inhibit third party innovations, particularly given the long term of copyright. Cf. Bonito Boats, Inc. v. Thunder Craft Boats, Inc., 489 U.S. 141, 146, 103 L. Ed. 2d 118, 131, 109 S.Ct. 971, 975 (1989) ("imitation and refinement through imitation are both necessary to invention itself and the very lifeblood of a competitive economy."). If the interface was copyrightable, the interface owner would also "lock up" an installed base of users accustomed to that interface.

C. Computer Languages Should Not Be Copyrightable

1. Overview of Computer Languages

This portion of the brief discusses the second way a user uses the Lotus menu words, as a computer language to write macro programs.n4 The Copyright Office has defined a computer language as "a programming language used by a programmer for writing a computer program."n5 A commentator similarly has described a computer language as "a code that both the computer operator and the computer understand. The operator uses the language to express what is to be done, and the computer understands the language and performs the desired actions."n6

A computer language thus is similar in many respects to an ordinary language such as English. Like an ordinary language, it

consists of basic building blocks of communication. It provides rules (grammar and syntax) for the combination of words or terms to form expressive works that can be understood by their authors and other programmers.

Unlike ordinary languages, computer languages serve an important additional purpose. They not only permit communication between different computer programmers, but also allow programmers to communicate with a computer and tell it what to do. They are highly utilitarian languages that tell the computer how to implement the desired functions, operations, and results desired by the program. In short, they permit computers to "talk" to each other and to exchange that which will be understood by the operators of the computer.

A computer language is fundamentally different than a program written in that language. The rules for the language consist of definitions and syntax, usually written in alphabetical or some other logical order. However, a program uses selected words of the language in the order appropriate for the function being performed. This concept is illustrated by the following example using the BASIC programming language. (The commands are listed in the alphabetical order used in the BASIC manual, and are only the small group of the many available BASIC commands which are actually used in the sample program, plus a few others.)

COMPARISON OF BASIC COMMANDS
WITH A PROGRAM WRITTEN IN BASIC

BASIC Commands	Sample Program
DATA	10 PRINT "This program adds."
END	20 PRINT "Type a number."
GOTO	30 INPUT A
IF/THEN	40 PRINT "Type another number."
INPUT	50 INPUT B
LET	60 LET C = A + B
PRINT	70 PRINT "The sum is", C
READ	80 END

The above program asks the user to type in two numbers; it then computes the sum of the two numbers and prints the sum. In the terminology of 17 U.S.C. sect. 101, the eight lines numbered 10 through 80 are the "set of statements or instructions"

of the program; the printed sum is the "certain result" of the program. Thus, the program on the right is different from the list of commands on the left.

2. The Lotus Macro Language

In this case, the Lotus 1-2-3 menu command words and the sequencing or syntax of those words define a computer language. Lotus 1-2-3 allows users to write their own computer programs, called "macros," using this programming language. See Lotus v. Borland, 49 F.3d at 809, Pet. App. at 4a. The District Court defined a "macro language" as "a feature by which a user may define a very short keystroke sequence as equivalent to a longer keystroke sequence."n7 As explained by the District Court, the Lotus 1-2-3 macro language permits a user to write a series of executable computer operations (the "macros") by using a language of special abbreviated programming commands (such as "R", "F", or "C"), and special symbols (such as "/" or "{?}"). Borland IV, 831 F.Supp. at 226-27, Pet. App. at 31a-32a. Thus, the 1-2-3 macro language is a set of rules for writing a computer program (the macro), and meets the definition of a computer language discussed above. Once users become reasonably adept with the operation of the spreadsheet program, they will often write macros to save time and effort.

Lotus has argued that the menu command hierarchy is a "computer program," and hence protectible under the copyright laws. For the reasons discussed above, this is not so. The words and syntax of the hierarchy can be used to write copyrightable computer programs, i.e. the macros. However, when the Lotus menu words and syntax are used in this manner, they are a language, not a program. They are the words from which the macro programmer creates the macros, which are the "set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result" under 17 U.S.C. sect. 101. In this respect, the Lotus menu words and syntax are similar to the words and syntax of a "national language," like English or French, which can be used to create copyrightable works of literature, and are similar to the BASIC language example noted above.

Again, this can be illustrated by an example comparing some of the Lotus command words with a macro written using those words. In the following example, the left column lists the "File" and "Data" commands used in the first two lines of the sample macro, in the order in which they appear in Lotus 1-2-3,

as shown at J.A. 900-906. The macro example on the right is taken from the Declaration of Judith S. Olson, J.A. 508-09.

COMPARISON OF LOTUS 1-2-3 COMMANDS
WITH A LOTUS MACRO PROGRAM

Lotus Commands	Sample Macro
"/F"ile Commands	/fit{?}~
. . .	/dprfcfe{?}~
"I"mport	i.{end}{down}~
"T"ext	o{right}~g
. . .	
"/D"ata Commands	
. . .	
"P"arse	
"F"ormat Line	
"C"reate	
"E"dit	
. . .	
"R"eset	

As Dr. Olson's declaration explains (J.A. 508-09), the macro program is used to extract data from another file, such as a database, and re-format it for use in a spreadsheet. The first line of the macro, for instance, opens a file ("f"), and imports it ("i") from the database as text ("t"). (Unlike BASIC, which as shown in the previous example uses entire English-looking words such as "PRINT," the Lotus macro language often uses just the first letter of the relevant command. See Borland IV, 831 F.Supp. at 226-27, Pet. App. at 31a.) The four lines of the macro are the "set of statements or instructions" of the program; the re-formatted data file is the "certain result" of the program. As was the case with the previous example, the program on the right is different from the list of commands on the left.

3. Reasons Why Copyright Protection Is Not Appropriate

It is very important to distinguish a copyrightable computer program from the uncopyrightable language in which the program is written. Copyright law was intended to cover the expressive nature of an author's works, and to encourage their creation.

However, it was never meant to cover the basic building blocks, or language, used to create the works.n8 Protecting the expression in a work is quite different from protecting the building blocks used to create the work. Protecting a novel written in the English language is not the same as protecting the English language itself.

Nor is protecting a work of art such as a painting the same as protecting the various colors of paint used to create the painting.

As the above examples make clear, a computer language is not the same as the "computer program" defined in 17 U.S.C. sect. 101. Accordingly, before this case, no one would have seriously argued that any language could be copyrightable, let alone a computer language.n9

There also are a number of good policy reasons why computer languages should not be copyrightable. The reasons discussed above why copyright protection is not appropriate for human/computer interfaces generally also apply to computer languages. Standard languages are very important for a wide variety of reasons: promoting ease of use by programmers, avoiding lock-ins of the user base and the technology by the first company on the market to write a language, providing incentives to create new works using a standard language, and preventing the disincentives that arise from requiring each competitor to create its own language.

The First Circuit acknowledged the realities faced by the computer industry. The majority opinion recognized the importance of program compatibility and the importance of allowing users to run programs such as macros on different platforms. The majority opinion concluded that "forcing the user to cause the computer to perform the same operation in a different way ignores Congress"s direction in sect. 102(b) that 'methods of operation' are not copyrightable." Lotus v. Borland, 49 F.3d at 817-18, Pet. App. at 20a. Judge Boudin"s concurring opinion also correctly observed how providing a monopoly of a computer command structure would make the users "captives" of the first provider of the program, not because of any creative effort or investment by that programmer, but because of the "investment in learning made by the users." Id. at 821, Pet. App. at 26a-27a.

Providing copyright protection for a computer language, such as the Lotus macro language, would have an additional undesirable effect. Such protection would extend far past the computer language itself. For example, granting copyright

coverage for computer languages would allow the language owners to claim ownership of all programs written in the languages as derivative works. If languages were copyrightable, works created using the language would be derivative works of the language.n10 In that event, the computer language owner could argue that he owned all programs based on his language.n11 This objectionable result is a natural consequence of confusing the expressive work itself with the language used in creating the expressive work, and permitting copyright coverage for both..

D. The Menu Command Words Need Not Be Spelled Out in the Code

Finally, as computer scientists we wish to respond to one point in the Lotus brief which we believe to be incorrect or potentially misleading. Lotus states that the actual menu command words are "spelled out, in text, in the program code." Lotus Br. at 7. While the record cites at this point are to a Borland data file which needed to recognize the 1-2-3 command words and order to operate, Lotus' statement incorrectly implies that the words did in fact appear in the Lotus code. Since the Lotus code was never introduced into evidence, J.A. 300, one cannot conclude that the menu words actually appeared in the Lotus code. From a programming standpoint, there are a number of ways the Lotus code could have implemented the menu functions without spelling out the words in the code.

For example, an extremely common programming technique used for programs with international sales (such as word processors or database programs) does not spell out the menu command words in the program code. The menu words are instead contained in a data file which is separate from the code used to run the program. This makes the menu words easy to change when the program is sold to non-English speaking users, since the words are isolated in one location instead of being scattered throughout thousands or millions of lines of code. It also has the advantage of permitting users to customize the menu command words if they want to. While we do not know if this or a similar technique was actually used in Lotus 1-2-3, since the Lotus code was never placed into evidence it cannot be assumed that such a technique was not used. Thus, one cannot state, as does Lotus, that its menu words were "spelled out" in its code.

IV. CONCLUSION

As computer scientists, we believe that the majority and concurring opinions of the First Circuit correctly recognized the danger in extending copyright protection to the specific type of human/computer interfaces and computer languages at issue in this case. For people in our computerized society, it is vitally important to be able to communicate freely with computers, without threat of litigation. Computer interfaces and languages are meant for everyone to use. They were never meant to enable their creator to monopolize methods of operating computer-based machines. The District Court's decisions lost sight of the importance of being able to use common, standard computer interfaces. The First Circuit correctly understood the issues involved here, and remedied the District Court's error.

For the foregoing reasons, the judgment of the First Circuit should be affirmed.

Respectfully submitted,

Ron Kilgard
(Counsel of Record)
Karl M. Tilleman
Dalton, Gotto, Samson & Kilgard, P.L.C.
Suite 900
3101 North Central
Phoenix, AZ 85012
Phone: (602) 248-0088

Attorneys for Amici
Listed in Appendix A

December 1995

APPENDIX A
LIST OF SIGNATORIES TO THE BRIEF
[TO BE SUPPLIED]

******FOOTNOTES******

1 A full list of amici and their professional qualifications and contributions to the industry follows the signature page of this brief. Amici include computer scientists who developed or contributed to the development of (1) computer languages such as Ada, Lisp and Logo; (2) artificial intelligence; (3) robotics; (4) data compression; and (5) other important areas of computer science.

2 Sources for this background material include: Computer Associates Intern., Inc. v. Altai, Inc., 775 F.Supp. 544, 549-551 (E.D.N.Y. 1991), aff"d., 982 F.2d 693 (2d Cir. 1992); F. Friedman, E. Koffman, Problem Solving, Abstraction, and Design Using C++, at 6-18, (Addison-Wesley Publishing Co. 1994), ISBN 0-201-52649-2; IBM Dictionary of Computing (McGraw-Hill 10th Ed. 1993), ISBN 0-07-031488-8(HC) (hereinafter IBM Dictionary) ; Microsoft Press Computer Dictionary, (2d Ed. 1994), ISBN 1-55615-597-2 (hereinafter Microsoft Dictionary).

3 In addition to IBM, the companies include at least Dragon Systems, Inc. of Newton, Massachusetts, and Kurzweil Applied Intelligence, Inc. of Waltham, Massachusetts. See Wayne Rash, Jr., Talk Show: Voice-recognition Technology for the PC Lets Your Voice Run the Show, PC Magazine, December 20, 1994, at 203, 205; Richard A. Shaffer, Computers With Ears, Forbes, September 12, 1994, at 238.

4 Lotus has denominated the Lotus macro language as the method by which the macro programs can interact with the Lotus 1-2-3 program. See Declaration of Vern L. Raburn, para. 18 and Exhibit B, J.A. 530, 535. In this respect the Lotus macro language acts as a program-to-program interface specification. We understand that other amici are discussing the uncopyrightability of program-to-program interfaces, and therefore will not discuss it in this brief.

5 Compendium II of Copyright Office Practices, the Library of Congress, sect.326 at 300-32 (1984).

6 E. Lowry, Copyright Protection for Computer Languages: Creative Incentive or Technological Threat?, 39 Emory L.J. 1293, 1298 (1990) (citation omitted) (hereinafter "Lowry"). Another

commentator summarized a number of definitions of computer language as follows:

> A computer programming language is a formal system of expression including:
> (1) a set of vocabulary elements;
> (2) a set of syntax rules for combining vocabulary elements into statements; and
> (3) the assignment of meaning to statements that properly combine vocabulary elements in accordance with the syntax rules.

R. Stern, Copyright in Computer Programming Languages, 17 Rutgers Computer & Tech. L.J. 321, 327 (1991) (hereinafter "Stern").

7 See Lotus Development Corp. v. Borland International, Inc., 799 F.Supp. 203, 206 (D.Mass. 1992), Pet. App. at 110a. (hereinafter "Borland II"). This brief also refers to four other relevant opinions by the District Court: Lotus Development Corp. v. Paperback Software International, 740 F.Supp. 37 (D.Mass. 1990) (hereinafter "Paperback"); Lotus Development Corp. v. Borland International, Inc., 778 F.Supp. 78 (D.Mass. 1992) (hereinafter "Borland I"); Lotus Development Corp. v. Borland International, Inc., 831 F.Supp. 202 (D.Mass. 1993) (hereinafter "Borland III"); Lotus Development Corp. v. Borland International, Inc., 831 F.Supp. 223 (D.Mass. 1993) (hereinafter "Borland IV").

8 Lowry at 1315; I. Paul Goldstein, Copyright Principles, Law and Practice, sect.1.2.2.4 at 16 (1989). See also Stern at 349, 364.

9 The District Court implicitly granted copyright protection to the
Lotus menu command hierarchy as a computer language. Borland IV, 831 F.Supp. at 229-230, 234, Pet. App. at 37a-39a, 47a; see also Borland II, 799 F.Supp. at 213-14, Pet. App. at 122a-125a. The District Court seemed to think that languages generally were copyrightable. Paperback, 740 F.Supp. at 72, Pet. App. at 243a-244a. See Stern at 323-24, 330, noting that Paperback's statement to this effect was unprecedented. This was one of the reasons why the District Court"s opinions generated such controversy in the computer industry and why a number of us filed an amicus brief in the First Circuit criticizing the District Court.

10 See the definition of derivative work in 17 U.S.C. sect.101:

> A 'derivative work' is a work based upon one or more preexisting works, such as a translation, . . . abridgment, condensation, or any other form in which a work may be recast, transformed, or adapted. A work consisting of editorial revisions, annotations, elaborations or other modifications . . . is a 'derivative work'.

Under this statute, a program is a work "based upon" the preexisting language; the language is "recast, transformed or adapted" into the program, which in any event is an "elaboration or other modification" of the language.

11 See 17 U.S.C. sect.106(2).