No. 17-2145

United States Court of Appeals for the Federal Circuit

CISCO SYSTEMS, INC.,

Plaintiff-Appellant,

v.

ARISTA NETWORKS, INC.,

Defendant-Appellee.

On Appeal from the United States District Court
for the Northern District of California
No. 5:14-CV-05344, Hon. Beth Labson Freeman

BRIEF FOR *AMICI CURIAE* THE MATHWORKS, INC., SAS INSTITUTE INC., ADOBE SYSTEMS INC., SYNOPSYS, INC., ORACLE CORPORATION, AND SYMANTEC CORPORATION IN SUPPORT OF PLAINTIFF-APPELLANT

Peter K. Stris
Elizabeth Rogers Brannen
Principal Counsel
Dana Berkowitz
Victor O'Connell
STRIS & MAHER LLP
725 S. Figueroa St., Ste. 1830
Los Angeles, CA 90017
Telephone: (213) 995-6800
Facsimile: (213) 261-0299
elizabeth.brannen@strismaher.com

Counsel for *Amici Curiae*

CERTIFICATE OF INTEREST

Pursuant to Federal Circuit Rules 29(a) and 47.4, counsel for *amici* certifies the following:

1. The full names of the *amici* represented by me are: The MathWorks, Inc., SAS Institute Inc., Adobe Systems Inc., Synopsys, Inc., Oracle Corporation, and Symantec Corporation.

2. The name of the real party in interest (if the party named in the caption is not the real party in interest) represented by me is: N/A.

3. All parent corporations and any publicly held companies that own 10 percent or more of the stock of the *amici curiae* represented by me are: The Vanguard Group owns 10 percent or more of Synopsys, Inc.'s common stock. None for the other *amici*.

4. The names of all law firms and the partners or associates that appeared for the party or *amici* now represented by me in the trial court or are expected to appear in this Court are: Elizabeth Rogers Brannen, Peter K. Stris, Dana Berkowitz, and Victor O'Connell, Stris & Maher LLP

Dated: September 20, 2017

/s/ Elizabeth Rogers Brannen
Elizabeth Rogers Brannen

Counsel for *Amici Curiae*

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Cases

Constitution

Statutes

Other Authorities

INTEREST AND IDENTITY OF *AMICI CURIAE*[1]

Amici are leading software companies. They devote significant resources to developing products in an industry that drives the national economy. Together, they employ tens of thousands of people in high quality jobs, including software engineering, customer support, and sales-related positions. *Amici* monetize their software products through licensing fees.

This case will influence the environment that *amici* face in their core business. Even for *amici* that offer combined hardware and software products, software serves as a vital differentiator in the market. *Amici* have a strong interest in a balanced application of copyright law—one that permits companies to use *ideas* in the marketplace to develop their software, but prevents competitors from taking the *expression* of those ideas that is the fruit of *amici*'s creativity. Their first-hand perspective will assist the Court in evaluating this appeal.

Adobe Systems, Inc. ("Adobe") is an American multinational computer software company headquartered in San Jose, California. For over 30 years, Adobe has been devoted to changing the world through digital experiences. Its flagship

[1] Stris & Maher LLP, counsel for *amici*, authored this brief. All parties consented to its filing. No person other than *amici* made a monetary contribution to the preparation or submission of this brief. Pursuant to Federal Circuit Rule 29(c)(5), no counsel for a party authored this brief in whole or in part, and no party or party's counsel contributed money intended to fund its preparation or submission.

products include Photoshop, Acrobat, Flash, and Adobe Marketing Cloud. Adobe provides the tools the world uses to create groundbreaking digital content, deploy it across all screens, measure and optimize it over time, and achieve greater business success.

Oracle Corporation ("Oracle") is a Delaware Corporation founded in 1977 and headquartered in Redwood Shores, California. Oracle provides products and services that address all aspects of corporate information technology. It develops and licenses a comprehensive line of enterprise software, including cloud-based and on-premise business applications, database, platform, and infrastructure solutions.

SAS Institute Inc. ("SAS") is a private software company that has for decades provided market-leading analytics software. Through innovative analytics, business intelligence and data management software and services, SAS helps customers at more than 83,000 sites make better decisions faster.

Symantec Corporation ("Symantec"), the world's leading cyber security company, helps organizations, governments and people secure their most important data wherever it lives. Organizations across the world look to Symantec for strategic, integrated solutions to defend against sophisticated attacks across endpoints, cloud and infrastructure. Likewise, a global community of more than 50 million people and families rely on Symantec's Norton and LifeLock product suites to protect their digital lives at home and across their devices.

Synopsys, Inc. ("Synopsys"), founded in 1986, is the world's 15th largest software company. It is a global leader in electronic design automation and semiconductor IP. Synopsys also provides innovative software security and quality solutions to companies developing the electronic products and software applications we rely on every day.

The MathWorks, Inc. ("MathWorks") is a medium-sized software company founded in 1984, with $850 million in annual revenues and more than 3,500 employees. MathWorks' flagship product is a computer program called MATLAB®, used by engineers and scientists to perform a vast range of numeric calculations and visualizations. MATLAB is used throughout the aerospace, defense, automotive, communications, and other industries, as well as essentially at all major universities worldwide, and has more than 2 million users worldwide. The cars people drive, the airplanes they fly in, and the smartphones they use contain algorithms and software that were designed by engineers using MATLAB.

INTRODUCTION AND SUMMARY OF ARGUMENT

The interpretation of the scènes à faire doctrine below threatens copyright protection for successful compilations everywhere, and especially in the computer software industry. The District Court's decision is at odds with the protection that Congress and this Court have accorded to computer programs. The District Court stripped Cisco of protection because Cisco's expression was clear and consistent, and had become popular. But these are hallmarks not only of the well-written software that *amici* strive to develop, but also of successful works of fiction, nonfiction, music, and film. By singling these characteristics out as reasons to *strip* protection, the decision below jeopardizes copyright protection for the best computer software and other compilations. It is not and should not be the law that the very qualities the Copyright Act seeks to promote undermine protection.

In this case, it is undisputed that Cisco devoted extensive resources to develop its command-line interface ("CLI"), which the jury concluded was original, copied, and infringed. It is similarly undisputed that others could (and did) implement the same underlying functionality without copying. Cisco Br. at 14-15 & n.2; *see also* Appx54378-54937 (HP Networking and Cisco CLI Reference Guide). But Arista chose to clone rather than to create. It took that unlawful shortcut precisely because the CLI developed by Cisco was successful, and it was faster and cheaper to copy Cisco's expression than to author an alternative that customers did not already know

and like. Cisco Br. at 17 (Appx46211 (Ullal) ("[I]t would take me 15 years and 15,000 engineers" to "compete with Cisco directly in the enterprise in a conventional way")). Instead of bearing the costs and risks of innovation, Arista simply offered Cisco's CLI at a deeply discounted price. Given that former Cisco executives founded Arista, perhaps this is not surprising. The Cisco CLI is what they knew. But it was not theirs to take.

The decision below punished Cisco precisely because Cisco's expression was well-written and popular. If this Court affirms, copyright protection is at risk whenever software is written clearly and consistently or becomes a premier choice of customers in an industry. That is exactly backward. Copyright law exists to incentivize successful expression, and Congress expressly extended its protections to original computer programs like Cisco's CLI. If affirmed, the law will punish *amici* and others for precisely the conduct that the Copyright Act seeks to promote, and will reward conduct the Act outlaws.

ARGUMENT

I. It Is Well-Settled, And For Good Reason, That Original Computer Programs Are Entitled To Copyright Protection.

The CLI commands at issue in this case fall squarely within the statutory definition of computer programs. *See* 17 U.S.C. § 101. They are "statements or instructions" used to "bring about a certain result" in routers and switches running Cisco's IOS operating system programs. *Id.*; *see also Oracle Am., Inc. v. Google*

Inc., 750 F.3d 1339, 1355-56 (Fed. Cir. 2014) (collecting authority that the structure, sequence, and organization of software programs as well as computer user interfaces fall squarely within the Copyright Act's definition of "computer programs" (citing, *inter alia*, *Johnson Controls, Inc. v. Phoenix Control Sys., Inc.*, 886 F.2d 1173, 1175 (9th Cir. 1989), *implied overruling on other grounds recognized by Perfect 10, Inc. v. Google, Inc.*, 653 F.3d 976 (9th Cir. 2011))).[2] The decision below is an end-run around this important protection.

User interfaces such as Cisco's CLI, as well as code written in computer languages, reflect many expressive choices. The author must decide which instructions to include, how to express each one, and how to organize and arrange them. Copyright protection for such expressive choices is essential to "promote the Progress of Science. . ." in the software industry. U.S. Const., art. I, § 8, cl. 8.

A. Computer programs are copyrightable.

Computer programs are inherently functional. But they are also undisputedly "literary works" eligible for copyright protection. *Oracle*, 750 F.3d at 1354. The

[2] As Cisco's brief explains, the trial in this case concerned Arista's alleged infringement of Cisco's copyrights in four operating system programs collectively known as "IOS." Cisco Br. at 6. The user interfaces of Cisco's IOS are the works at issue. *Id.* These interfaces are text-based. *Id.* They permit users to communicate with Cisco network switches and routers running the IOS operating systems using words and phrases that Cisco developed, and are known as command-line interfaces, or "CLIs." *Id.*

Copyright Act defines them as follows: "[a] 'computer program' is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." 17 U.S.C. § 101.

Congress did not include this definition by happenstance. Extensive discussions in the 1970s and 1980s about intellectual property protection for software culminated in general acceptance, worldwide, that computer programs should receive copyright protection. *See Copyright Protection of Computer Software*, World Intellectual Property Organization, http://www.wipo.int/copyright/en/activities/software.html (last visited Sept. 20, 2017). In 1980, Congress added the definition of "computer program" recited above to eliminate any doubt about whether such programs are protectable. *See* H.R. Rep. No. 96-1307, pt. 2, at 19 (1980), *reprinted in* 1980 U.S.C.C.A.N. 6460 (noting that adding this definition "has the effect of clearly applying the 1976 law to computer programs").[3]

Extending copyright protection to computer programs makes good sense because, like virtually all written communication, an original program embodies a number of significant choices—not only about what functionality to include at a

[3] By April 1994, all members of the General Agreement on Tariffs and Trade had "committed, as a matter of international trade policy, to the protection of computer programs by copyright law." Jane C. Ginsburg, *Four Reasons and a Paradox: The Manifest Superiority of Copyright over Sui Generis Protection of Computer Software*, 94 Colum. L. Rev. 2559, 2562 (1994).

conceptual level, but about how to express the statements and instructions that together comprise the resulting work. That is true in at least two respects:

First, the author must decide how to express each individual function or instruction included in the program. Even if (and sometimes especially if) a command is concise, the decision about how to express it may reflect creativity.[4]

Consider MathWorks' *tic*/*toc* command. MathWorks' flagship MATLAB® program allows users to determine the amount of time it will take to execute certain code by writing "*tic*" at the beginning of the code they want to time and "*toc*" at the end. Anyone is free (assuming no patent violation) to create a program or computer script to perform that function, but there are countless ways to express it without using the words "*tic*" and "*toc*." Alternatives might include reference to *time* or *timer*, *clock*, *start*, *stop*, *begin*, *end*, and *elapsed*, to name a few. SAS uses "STIMER" and "FULLSTIMER" to designate different execution timers, and "STIMEFMT" to specify timer format.

[4] Words matter. In other contexts, the importance of diction is more intuitive. *See, e.g.*, *Towne v. Eisner*, 245 U.S. 418, 425 (1918) (Holmes, J.) ("A word is not a crystal, transparent and unchanged, it is the skin of a living thought and may vary greatly in color and content according to the circumstances and the time in which it is used."); *Cohen v. California*, 403 U.S. 15, 25 (1971) ("[W]hile the particular four-letter word being litigated here is perhaps more distasteful than most others of its genre, it is nevertheless often true that one man's vulgarity is another's lyric.").

While *tic/toc* is clever and fun, the modicum of originality required for copyright protection is lower. It is "extremely low." *Feist Publ'ns, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 345 (1991). As this Court has recognized: if the commands used to express instructions are original and could have been expressed differently, they are entitled to protection. *Oracle*, 750 F.3d at 1366-67.[5]

Here are just a few of the things *amici* may consider in deciding how to express any given underlying function of a program:

- Is the name descriptive?
- Is it easy to learn and remember?
- Is it an appropriate length (short enough such that it is not too burdensome to type but not too short to convey anything informative)?
- Is there a particular reason why it should be long or short given the context?
- Is it aesthetically pleasing, or even humorous?
- Should it be case-sensitive?
- Should it be expressed in all capitals, lower case, or mixed capitalization?
- Should it require punctuation? If so, which punctuation (commas, semi-colons, brackets, curly braces, etc.)?

Using their business and technical expertise, authors make these and other judgments to write individual instructions that will appeal to their target audience.

Second, authors exercise choice and may bring great creativity to deciding how to structure and arrange the various instructions that comprise their original

[5] The District Court determined that only various *compilations* of Cisco commands were entitled to copyright protection. That determination is not the subject of this appeal. Accordingly, this appeal does not implicate questions about whether short phrases or statements are entitled to copyright protection. *Amici* express no view about whether the District Court's analysis on this issue was correct.

programs. Most commercial software programs involve multiple instructions. The CLI at issue here, for example, involved more than 500 multiword command expressions that Arista copied, nearly verbatim. Cisco Br. at 16 (citing Appx1857-2067 (comparing commands)). Cisco decided not only how to express each one, but how to organize and arrange them into its particular program structure and organization.[6] For example, Cisco's "show" commands include a subset for "show ipv6." Appx2005-2011. "[S]how ipv6," in turn, includes a number of additional commands that Cisco chose not only to phrase in a certain way, but to include and place here, as opposed to elsewhere within its arrangement. *Id.*

Even if no single instruction or command were protectable in isolation, the instructions that collectively comprise an original computer program are expressive compilations or collections. And the expressive choices associated with developing original software *increase* with the complexity and demands of a project. For this reason, adding functionality tends to expand the manner in which an author may

[6] That the CLI enables users to bring about a certain result using declarative sentences humans can read and understand left Cisco even more choices and flexibility, and therefore more room for originality and creativity, than programmers may enjoy when creating works in a specific computer-to-computer coding language. *See* Cisco Br. at 5-6. This is particularly true here because Cisco was the original creator of the underlying hardware systems, which had not previously been offered. *Id.* at 6-7, 11. That makes the ramifications of the decision below even more troubling and potentially harmful to the software industry.

express, organize, and arrange all of a program's constituent instructions, and can expand the author's choices exponentially.

The way *amici* express the functionality of their programs accounts for much of the value that makes their products (and their customers) successful. Different customer bases have different backgrounds and levels of technical expertise, and even people within a target audience may respond differently to different expressions. A program expressed in a way that appeals to its target audience will become popular. A program expressed unappealingly will struggle and perhaps become obsolete or never take hold. This is not just a question of functionality; good software is high-quality expression. It reflects judgment and artistry in its creation.

B. Copyright protection is essential to promote the progress of science and useful arts in the software industry.

The software industry plays a vital role in the American economy.[7] Most software companies rely on the revenue generated from licensing fees to run their businesses. That model is viable only if *amici* and their peers can prevent competitors and other non-customers from reaping the benefits of their software without paying. *Amici* all rely on copyright to help protect their computer programs

[7] According to a study the BSA conducted in 2016, in 2014 the software industry contributed 2.5 million direct and 9.8 million total jobs to the U.S. economy, and in 2015 a software developer's annual wage was more than twice the average annual wage for all U.S. occupations. *The $1 Trillion Economic Impact of Software* at 1, BSA, The Software Alliance (June 2016), http://softwareimpact.bsa.org/pdf/Economic_Impact_of_Software_Report.pdf ("BSA Study").

from copying and to receive licensing fees, which allows them to invest back into creating more innovative and improved products.[8] Without adequate copyright protection, the industry would collapse.

To be sure, copyright is not the only form of intellectual property protection available for software. Patents may protect innovative aspects of a program's functionality. But only copyright law ensures that companies receive intellectual property protection for programs implementing functionality that may not be new and non-obvious, but nevertheless embody valuable *original expression* that Congress has decided should be incentivized. Without copyright, nothing would prevent competitors from stealing many computer programs and undercutting their authors with deeply discounted imitations.

Copyright protection thus provides a crucial incentive for companies to develop new and better programs in the first place. Doing so requires enormous resources. A single program can take months or even years, and require the investment of millions of dollars.[9] Permitting free and brazen appropriation of

---

[8] While software companies are free to choose any business model, even those that license their software for free rely upon copyright protection and cannot simply assimilate the work of others who do not use the same business model.

[9] For example, in November 2016, pursuant to its Exascale Computing Project, the U.S. Department of Energy awarded $34 million to 25 recipients whose 35 proposals for software development were selected for funding. *The Exascale Computing Project Awards $34 Million for Software Development*, Los Alamos National Laboratory (Nov. 10, 2016), http://www.lanl.gov/discover/news-release-

computer programs would eviscerate the incentive to create them: if competitors are free to plagiarize and then flood the market with "drop-in replacements" offered at half-price, there is no reason for companies to continue dedicating the resources necessary to develop new programs, and every reason for them to take the fast and easy route of imitation.[10]

As a result, this Court should reject arguments and decisions, such as the one below, that would permit wholesale copying of original computer programs, whether on the basis of scènes à faire or otherwise. Permitting slavish copying of drop-in replacements that share 99.999% similarity with original programs that another company went to great lengths to develop will undermine, not "promote the Progress of Science. . ." U.S. Const., art. I, § 8, cl. 8.

Defenses such as scènes à faire serve an important purpose: to ensure that authors do not enjoy a monopoly on *ideas* so that others remain free to use those

---

archive/2016/November/11.10-exascale-computing-software-development.php. According to the BSA, in 2014, software companies invested $52 billion in research and development, accounting for 17.2% of all domestic R&D in the United States. BSA Study at 1.

[10] *See also* John Villasenor, *Why Patents and Copyright Protections Are More Important Than Ever Before*, Scientific American (Nov. 14, 2013), https://www.scientificamerican.com/article/why-patents-copyright-protections-are-more-important-than-ever/ (observing that without patent and copyright protection, entrepreneurs in software and other industries "would risk having their inventions simply stolen" by competitors, which would make them "far less likely to launch . . . new companies, leaving us all without the benefit of their innovations").

ideas in developing other programs. *See, e.g.*, *Mitel, Inc. v. Iqtel, Inc.*, 124 F.3d 1366, 1375 (10th Cir. 1997) (citing *Gates Rubber Co. v. Bando Chem. Indus., Ltd.*, 9 F.3d 823, 838 (10th Cir. 1993); *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 711 (2d Cir. 1997)). To ensure that copyright law appropriately rewards and stimulates creativity in future works, however, courts must not interpret defenses in a manner that effectively removes the statutory entitlement to copyright protection for original computer programs. Scènes à faire must not be interpreted so broadly that it obliterates an original author's important rights in its particular original *expression* of software compilations. Unfortunately, that happened below. If affirmed, that spells trouble for the software industry.

II. The Decision Below Endorsed An Unprecedented Expansion Of Scènes À Faire That Will Eliminate Copyright Protection For Successful Computer Compilations.

As Cisco's brief explains, the Court instructed the jury that it could find five elements of Cisco's works to be protectable as compilations if it found them to be original: (1) the selection and arrangement of multiword command-line expressions; (2) the selection and arrangement of modes and prompts; (3) the collection of screen responses and outputs; (4) the collection of help descriptions; and (5) the user interfaces as a whole as compilations of these four elements. Cisco Br. at 19-20. The jury found infringement of at least one of these compilations (we do not know which). *Id.* at 21. But the jury also found, as to whichever compilation or

compilations it found infringement, that Arista had proven its affirmative defense of scènes à faire. *Id.*

Analyzing only the first of the five compilations the jury may have found to be infringed, the decision below concluded that substantial evidence supports the scènes à faire verdict. It did so on two grounds: (1) evidence of functionality as an external constraint, Appx9-11; and (2) evidence of consumer demand as an external constraint, Appx11-12.

That decision was fundamentally flawed—although the jury necessarily found that Arista infringed *particular original expression* in one or more of Cisco's compilations, both aspects of the District Court's scènes à faire analysis were untethered to that expression:

1. With respect to functionality, the District Court mistook requirements and desires about *what* functionality to include for substantial evidence of external constraints on *how to express* that functionality. It also credited the fallacy that if specific sub-portions were externally constrained (a premise *amici* do not believe any evidence established), the jury had an evidentiary basis to conclude that so too was the entire compilation.

2. With respect to consumer demand, the District Court failed to heed its own statement of the law recognizing that scènes à faire must exist at the time of

original creation. Instead, it allowed the jury to make Cisco a victim of its own success.

In short, there was no substantial evidence of scènes à faire as to Cisco's *compilations*. The logic of the decision below would eliminate copyright protection for computer programs. These legal errors must not take hold.

A. The District Court erred in finding evidence sufficient to support scènes à faire on the basis of functionality as an external constraint.

The first ground on which the District Court denied JMOL was its conclusion that "[t]he jury could reasonably infer that constraints flowing from the overall industry context and the basic functional nature of the CLI dictated the overall structure and arrangement of Cisco's asserted compilation of commands that the jury found was original and infringed." Appx11. The record lacks substantial evidence for this inference. In concluding otherwise, the decision below misconstrued scènes à faire in a manner that unduly expands the doctrine and renders it at odds with the purposes of the Copyright Act.

1. The District Court's analysis of functionality as a basis for scènes à faire cannot be reconciled with the jury verdict as a matter of logic. The infringement verdict *required* the jury to find that at least one of the compilations resulted from Cisco's original selection, coordination, and/or arrangement of other elements that need not themselves be original or individually protectable. *See Feist Publ'ns, Inc.*, 499 U.S. at 349; *see also M. Kramer Mfg. Co. v. Andrews*, 783 F.2d 421, 440 (4th

Cir. 1986) ("[T]he validity of a copyright of a compilation or derivative work depends on the originality of the compiler's individual contribution to the work or material regardless of whether the individual items in the material have been or ever could have been subject to copyright."). In finding infringement, the jury necessarily found that at least one of the five *compilations* was protectable expression.

The jury also rejected Arista's merger defense. This means the jury necessarily reached the indisputably correct conclusion that there were viable alternative ways to express the functionality of Cisco's compilations.[11] The District Court's conclusion that "there is evidence that at least certain selection and arrangement of multiword command-line arrangements were constrained by functionality, and preexisting network industry protocols," Appx9, is at odds with this verdict.

Merger and scènes à faire are closely related. *Oracle*, 750 F.3d at 1363. To the extent scènes à faire could exist absent merger, evidence would have to exist *at the compilation level* that Cisco's selection, collection, and arrangement of its numerous commands was "so commonplace" in the industry *when Cisco originally wrote them* that it would be appropriate to treat them like ideas. *Id.* ("In the computer context, the [scènes à faire] doctrine denies protection to program elements that are dictated

[11] Substantial evidence established that this was the case when Cisco created its compilations (the relevant time for scènes à faire), Cisco Br. at 7, and that it remained true when other competitors, such as HP, created CLIs implementing similar functionality using different expression, *see infra* at 20.

by external factors such as the mechanical specifications of the computer on which a particular program is intended to run or widely accepted programming practices within the computer industry." (citation and internal quotation marks omitted)). There was no evidence from which a reasonable jury could have reached this conclusion about the expression of Cisco's compilations. *Cf. Oracle*, 750 F.3d at 1364 (noting the absence of evidence that "the groupings and code chosen for the 37 Java API packages were driven by external factors or premised on features that were either commonplace or essential to the idea being expressed").

2. The District Court's discussion of functionality as an external constraint conflated the functionality of Cisco's CLI with the expression of that functionality. There is a fundamental difference between the two. The expression is copyrightable; the underlying idea of performing a specific function is not. *See, e.g.*, *Oracle*, 750 F.3d at 1354-55 (explaining the origin and basis for the "idea/expression dichotomy" and that, as a result, "those elements of a computer program that are necessarily incidental to its function are . . . unprotectable") (citations omitted).

To illustrate: Cisco chose to include commands relating to ipv6, the Internet's so-called "next generation protocol," and to group those commands in various different places in its hierarchical arrangement.[12] As with the other industry standard

[12] *IPV6 – The Next Generation Internet* (Feb. 10, 2006), https://www.ipv6.com/general/ipv6-the-next-generation-internet/.

protocols the District Court discussed (e.g., IGMP and OSPF), the decision to include ipv6-related commands for routers and switches capable of communicating using that standard is separate from *how to express* them and where to place them in the arrangement that Cisco selected for its CLI. As even the following limited examples reflect, Arista copied the expression and organization verbatim:

Cisco CLI Command Expression	**Arista CLI Command Expression**
clear ipv6 neighbors	clear ipv6 neighbors
clear ipv6 ospf force-spf	clear ipv6 ospf force-spf

Appx1874-1855; *see also* Appx51350.

Cisco CLI Command Expression	**Arista CLI Command Expression**
show ipv6 access-list	show ipv6 access-list
show ipv6 bgp	show ipv6 bgp
show ipv6 bgp community	show ipv6 bgp community
show ipv6 bgp neighbors	show ipv6 bgp neighbors
show ipv6 bgp summary	show ipv6 bgp summary
show ipv6 interface	show ipv6 interface
show ipv6 neighbors	show ipv6 neighbors
show ipv6 ospf	show ipv6 ospf
show ipv6 ospf border- routers	show ipv6 ospf border- routers
show ipv6 ospf interface	show ipv6 ospf interface
show ipv6 ospf neighbor	show ipv6 ospf neighbor
show ipv6 prefix-list	show ipv6 prefix-list
show ipv6 route	show ipv6 route
show ipv6 route summary	show ipv6 route summary
show ipv6 route tag	show ipv6 route tag

Appx2005-2011; *see also* Appx51356-51357.

External constraints did not compel Cisco's expression (or Arista's verbatim copying) of 500+ commands. The record makes clear that other competitors did not perceive Cisco's CLI commands, including their phrasing and organization, as

scenes that "must be done." Indeed, the HP Networking and Cisco CLI Reference Guide is replete with side-by-side comparisons of functionality expressed differently in HP's ProVision and Comware systems. Appx54378-54937. To protect a local password, for example, the three systems use the following commands, respectively:

ProVision	**Comware**	**Cisco**
ProVision(config) # no front-panel-security password-clear	<Comware>undo startup bootrom-access enable	Cisco(config) #no service password-recovery

Appx54435. To request information about the system and environment, they use the following:

ProVision	**Comware**	**Cisco**
ProVision# show system information ProVision# show modules	<Comware>display device manuinfo <Comware>display device verbose	Cisco#show inventory Cisco#show version

Appx54402.

The evidence of training further underscores the difference between the functionality of Cisco's CLIs and the expression of that function. Cisco offered extensive customer training so that network engineers could master the Cisco CLI. Cisco Br. at 10. Network engineers are familiar with the basic functions of network equipment such as devices and routers. If functionality dictated the expression, there would be little to learn beyond that. Arista recognized that was not the case for Cisco's CLI, touting that by copying, it was able to appeal to the large pool of

engineers already trained to use it. *See, e.g.*, Appx135-137; Appx45468-45470 (Tr. Ex. 197 (Arista e-mail chain)). What it was leveraging was a large population familiar with Cisco's specific expressions and compilations.

Because computer programs are inherently functional, courts must take special care in copyright cases to analyze and ensure that evidence concerns the expression of that functionality, not the idea of the functionality. The decision below falls prey to this fundamental error. It equated the choice of which functionality to implement with a functional constraint on how to express that functionality. The District Court noted, for example, that Arista's expert testified that "as a technical matter, the functional choice of features to be implemented in a system dictates the contents of the compilation of CLI commands. . . ." Appx9. From this, the District Court reasoned (wrongly) that "a reasonable jury could conclude the selection of commands to create the compilation was constrained by functionality." *Id.* That conclusion simply does not follow.

Of course "it [would not] make sense" to have commands related to functionality a vendor does not offer. *Id.* But evidence that choosing which features to implement dictated when and where the author had to create commands to express the underlying feature or functionality says nothing about whether functionality constrained the *expression* of those commands. All computer programs are functional, and therefore at some point someone must decide which functionality to

implement. But it is well settled, as discussed above, that the original expression they embody is also copyrightable. *See also, e.g.*, Br. for United States as *Amicus Curiae* at 10, *Google, Inc. v. Oracle Am., Inc.*, 135 S. Ct. 2887 (2015) (No. 14-410) ("Despite the inherently functional character of all computer code, the Copyright Act makes clear that such code can be copyrightable."). If evidence of a decision to implement certain functionality is enough to constitute evidence of an external constraint sufficient to support scènes à faire as to specific expression, the defense will become an exception that swallows the rule.

3. The decision below reflects a fundamental misunderstanding of what constitutes an external constraint. The District Court referred to evidence that Cisco selected certain commands defined in industry standard publications and selected others to be reasonable and logical, and to make sense to network engineers. Appx10-11. But influences are not constraints. Nor are broad aspirations or even instructions to achieve high-quality expression that is clear, concise, logical, and user-friendly. Such directives nevertheless permit writing and arranging instructions in potentially infinite ways. Here, for example, Arista also developed a non-Cisco based Linux CLI that a significant portion of its customers used and that was not accused of infringement. *See* Cisco Br. at 15 (citing Appx10802 (Duda)).

Unlike influences and aspirations, functional external constraints are things that serve to dictate design and eliminate choice in how programming is expressed. The case law helps illustrate what that means.

Consider *Mitel, Inc. v. Iqtel, Inc.*, on which the decision below relied. 124 F.3d 1366 (10th Cir. 1997). In *Mitel*, the Tenth Circuit affirmed a decision denying a preliminary injunction on the basis of its conclusion that certain command code descriptions were not original enough to qualify for copyright protection, and that others were sufficiently original but nevertheless scènes à faire. *Id.* at 1373-75. The work at issue was "an instruction set of over sixty four-digit numeric command codes," created by Mitel for use in programming its call controllers.[13] *Id.* at 1368. The first three digits of the command code comprised the "register," which specified the line or route number, or a group of arbitrarily selected functions. *Id.* at 1368-69. The last digit comprised the "description;" "a number or symbol (usually 0 through 9, *, or #) that represent[ed] a particular setting within each function." *Id.* at 1369.[14]

[13] A call controller is "computer hardware that enhances the utility of a telephone system by automating the selection of a particular long distance carrier and activating optional features such as speed dialing." *Id.*

[14] In the baud-rate function, for example, selecting the description digit "1" would correspond to 110 baud, whereas selecting the description digit "4" would stand for 1200 baud. In other functions, the description digits would each stand for something else; for example, in the time-to-answer function, selecting the description digit "4" would correspond to 40 seconds. *Id.* at 1369.

The district court had concluded that nothing was sufficiently original to qualify for copyright protection. *Id.* at 1373.

On appeal, the Tenth Circuit affirmed based upon the different rationale that the "content of the 'values' created by Mitel and assigned to its 'descriptions'" were scènes à faire. *Id.* at 1373-76. The court cited evidence that Mitel selected many of the description digit values based upon customer demand "to ensure compatibility with equipment already installed in" its customer's offices; that it frequently simply matched them in equally divided ascending steps; that Mitel had determined many of them pursuant to "[s]tandard programming conventions such as '1' for 'on' and '0' for 'off';" and that they were "dictated by the limits inherent in the public telephone networks that the call controllers accessed." *Id.* at 1375. From these specific findings, the Tenth Circuit concluded that "although Mitel's values constitute non-arbitrary original expression, they are unprotectable as [scènes à faire] because they were dictated by external functionality and compatibility requirements of the computer and telecommunications industries." *Id.* at 1375-76.

Setting aside whether *Mitel* was correctly decided, the contrast between the nature and quantum of evidence supporting the Tenth Circuit's scènes à faire determination and the decision below is revealing. The record evidence in *Mitel* did not merely speak to high-level aspirations to please customers or implement functionality in a manner consistent with existing industry practices. The Tenth

Circuit identified and relied upon evidence that Mitel's expression of the values corresponding to the last digit in its four-digit command codes was actually constrained by specific requirements, equipment, and basic coding conventions. The decision below drew an analogy to *Mitel*—without identifying evidence that Cisco had to express any command, let alone the set of commands in its CLI compilations, in the way that it did to ensure compatibility with existing equipment, to comply with industry standards, or to adhere to specific programming practices akin to always using "1" for "on" and "0" for "off."

Nothing showed that the commands Arista copied were "as a practical matter indispensable," or even "standard, in the treatment of a given idea" at the time Cisco created them. *Apple Computer, Inc. v. Microsoft Corp.*, 35 F.3d 1435, 1444 (9th Cir. 1994) (citing *Frybarger v. Int'l Bus. Machs. Corp.*, 812 F.2d 525, 530 (9th Cir. 1987); quoting *Atari, Inc. v. N. Am. Philips Consumer Elecs. Corp.*, 672 F.2d 607, 616 (7th Cir. 1982)) (internal quotation marks omitted). At most the decision below describes substantial evidence that industry considerations influenced *what* functionality Cisco selected to express, and that Cisco grouped the expression of certain functionality in one (but not the only) logical manner. That is simply not evidence that external constraints determined the manner in which Cisco *expressed* any of the commands or compilations of commands it created.

4. Evidence that Cisco selected tiny *portions* of its CLI from external sources cannot support an inference that external constraints dictated the *precise expression of the compilations*. Nothing identified in the decision below constitutes substantial evidence that the compilations were dictated. The District Court erred in summarily reaching the opposite conclusion. Namely, that "a reasonable jury could infer from the evidence regarding *portions* of the compilation that the *entire* compilation was directed by external factors." Appx13-14 (emphasis added). No, a reasonable jury could not reach that conclusion.

Copyright law protects compilations of elements that are not themselves protectable. In *Apple Computer*, the Ninth Circuit "held that the basic *ideas* of a desktop metaphor in a computer's operating system—windows on the computer screen, icons representing familiar office objects, drop-down menus and objects that open and close—were not individually protectable." *Metcalf v. Bochco*, 294 F.3d 1069, 1074 (9th Cir. 2002) (citing 35 F.3d 1435, 1443-44 (9th Cir. 1994)) (emphasis added). As the Ninth Circuit explained in *Metcalf*, though: "[h]owever, consistent with *Shaw* [*v. Lindheim*, 919 F.2d 1353, 1356 (9th Cir. 1990)], we also held that infringement can be based on original selection and arrangement of unprotected elements," and concluded that "Apple was entitled to and did license the way in which it put unprotectable ideas together through the creative use of animation,

overlapping windows, and well-designed icons." *Id.* at 1074 (citations and internal quotation marks omitted).

Amici thus doubt whether evidence that certain portions of a compilation are unprotectable could ever suffice to prove that the entire compilation is unprotectable. *See Oracle*, 750 F.3d at 1368 (defendant, "like any author, is not permitted to employ the precise phrasing or precise structure chosen by Oracle to flesh out the substance of its packages—the details and arrangement of the prose").[15] But even assuming it might in some rare case, the portions as to which the District Court cited evidence did not implicate anything close to the 500+ commands in Cisco's multiword command compilation. *See* Cisco Br. at 48-49, n.8 (explaining that the examples discussed in the decision below were "vanishingly trivial").

This error goes to the heart of why affirming would be so damaging to the software industry and would undercut the very incentive to create that the Copyright Act exists to encourage. If others can usurp large (or even small, but valuable) portions of programs based upon evidence that external constraints dictated a small portion of the expression they contain, companies will not be able to count on meaningful copyright protection for their original programs. *See, e.g.*, *Harper &*

[15] *Apple Computer* shows that an improperly broad and incorrect interpretation of the scènes à faire doctrine would threaten to undermine copyright protection for graphical user interfaces ("GUIs") too. *Amici* urge this Court to avoid that.

Row, Publishers, Inc. v. Nation Enters., 471 U.S. 539, 539-49, 556-69 (1985) (reversing judgment of no liability where magazine had copied 300 words amounting to 13% of accused article from the unpublished memoirs of former President Ford). Second entrants would simply take whatever is popular from original authors, knowing that their behavior would likely be excused—even if the original work succeeds because of the quality of its expression and even if there are multiple other ways to express the compilation of functionality embodied in the program.

B. The District Court erred in finding evidence sufficient to support scènes à faire on the basis of customer demand as an external constraint.

The District Court also erred in its analysis of "customer demand" as an external constraint supporting the verdict of scènes à faire. In concluding that "there is also substantial evidence that selection and arrangement of the multiword command lines were constrained by customer demands," Appx11, the District Court repeated the same errors from its analysis of functionality as an external constraint, and made an additional error.

1. The repeat errors include confusing influences with constraints and failing to require evidence on Cisco's precise compilations as opposed to a few constituent elements. The District Court stated that "a reasonable jury could conclude that Cisco's selection and arrangement of elements in the compilation of

multiword command lines was directed by the need to satisfy customers who wanted consistency, as well as functions from pre-existing systems." Appx12. By this logic, the only expression that would be protected is expression that is inconsistent and difficult to understand and remember. That cannot be the law. Customers in *amici*'s industry want programs that are logical and easy to use, just as readers of literature want text that is expressive, easy to understand, and consistent within the general constraints of the specific plot or story, or original series. They want programs that reference terms and conventions with which they are familiar. If coherence eviscerates copyright protection, the results for the software industry are untenable. Being concise, logical, and consistent are goals of most authors. They are not, in fairness, external constraints on any given multiword command or on Cisco's compilations. To treat them as such sets up a foregone conclusion that could be used to render compilations unprotectable.

2. Cisco's internal decision to remain consistent with expression it created, and that customers liked, is not evidence of an external constraint, or of scènes à faire at the relevant time. Despite having correctly instructed the jury and recognized in its decision that for scènes à faire, the relevant time is the time of creation, not the time of infringement, *see* Appx8 (Instr. No. 61), the District Court's analysis was unfaithful to this requirement. Instead, its discussion of customer demand as a constraint supporting scènes à faire conflated time-of-creation with later

market penetration. The decision relies upon evidence of certain protocols that Cisco "started adding" *after* introducing its products. Appx12. *At the time the works were created*, however, there is no evidence of customer demand for the *expression* arranged into Cisco's compilations.

The rationale below distorted scènes à faire to excuse wholesale infringement of successful compilations because customers eventually came to appreciate and embrace the quality of the Cisco-created expression. At significant expense and risk, Cisco created groundbreaking content and a new market. Arista took the shortcut of copying Cisco's expression. The free ride afforded Arista a 99.999% clone that it touted as a "drop-in replacement" and offered at a fraction of the price. That is copying of the worst kind.

CONCLUSION

To ensure that copyright law continues to encourage the development of new and original computer programs in the software industry, *amici* respectfully urge this Court to reverse.

Dated: September 20, 2017

Respectfully submitted,

/s/ Elizabeth Rogers Brannen
Peter K. Stris
Elizabeth Rogers Brannen
Dana Berkowitz
Victor O'Connell
STRIS & MAHER LLP
725 S. Figueroa Street, Suite 1830
Los Angeles, CA 90017
Telephone: (213) 995-6800
Facsimile: (213) 261-0299
elizabeth.brannen@strismaher.com

Counsel for *Amici Curiae*

CERTIFICATE OF COMPLIANCE

I, Elizabeth Rogers Brannen, hereby certify that this brief complies with the type-volume limitation of Federal Rule of Appellate Procedure 29(a)(5) and Circuit Rule 28.1(b)(1)(A) as it contains 6,994 words, excluding the parts of the brief exempted by Federal Rule of Appellate Procedure 32(a)(7)(B)(iii). I further certify that this brief complies with the format, typeface, and type-style requirements of Federal Rules of Appellate Procedure 32(a)(4)-(6).

Dated: September 20, 2017

/s/ Elizabeth Rogers Brannen
Elizabeth Rogers Brannen

Counsel for *Amici Curiae*

CERTIFICATE OF SERVICE

I, Elizabeth Rogers Brannen, hereby certify that on September 20, 2017, I electronically filed the foregoing brief with the Clerk of the Court of the United States Court of Appeals for the Federal Circuit by using the appellate CM/ECF system. Participants who are registered CM/ECF users will be served by the appellate CM/ECF system.

Dated: September 20, 2017

/s/ Elizabeth Rogers Brannen
Elizabeth Rogers Brannen

Counsel for *Amici Curiae*