

J CLERK

JUL 13 2004

2004 JUL 12 P 11: 48

MARKUS B. ZIMMER, CLERK

DEPUTY CLERK

Brent O. Hatch (5715)
HATCH, JAMES & DODGE
10 West Broadway, Suite 400
Salt Lake City, Utah 84101
Telephone: (801) 363-6363
Facsimile: (801) 363-6666

Robert Silver (admitted pro hac vice)
BOIES, SCHILLER & FLEXNER LLP
333 Main Street
Armonk, New York 10504
Telephone: (914) 749-8200
Facsimile: (914) 749-8300

Stephen N. Zack (admitted pro hac vice)
Mark J. Heise (admitted pro hac vice)
BOIES, SCHILLER & FLEXNER LLP
Bank of America Tower – Suite 2800
100 Southeast Second Street
Miami, Florida 33131
Telephone: (305) 539-8400
Facsimile: (305) 539-1307
Attorneys for Plaintiff The SCO Group, Inc.

Frederick S. Frei (admitted pro hac vice)
Aldo Noto (admitted pro hac vice)
John K. Harrop (admitted pro hac vice)
ANDREWS KURTH LLP
1701 Pennsylvania Ave. NW, Suite 300
Washington, DC 20006
Telephone: (202) 662-2700
Facsimile: (202) 662-2739

**IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF UTAH**

THE SCO GROUP, INC.

Plaintiff/Counterclaim Defendant

vs.

INTERNATIONAL BUSINESS
MACHINES CORPORATION

Defendant/Counterclaim Plaintiff

**REPLY MEMORANDUM
REGARDING DISCOVERY**

Case No. 2:03-CV-0294 DAK

Honorable Dale A. Kimball
Magistrate Judge Brooke C. Wells

The SCO Group, Inc. ("SCO") through its respective counsel of record respectfully submits this Reply Memorandum Regarding Discovery.

205

Plaintiff The SCO Group, Inc. ("SCO") submits this Reply Memorandum in further support of its Memorandum dated May 28, 2004, filed pursuant to the Court's Order of March 3, 2004 (the "March Order"), identifying relevant discovery that SCO has requested, but has not received, from IBM.

PRELIMINARY STATEMENT

SCO here asks the Court to order IBM to produce the source code of the multiple versions of the AIX and Dynix/ptx computer operating systems and related documents that SCO has sought for over a year. These discovery requests remain at the heart of this litigation, because they are directly relevant to, and reasonably calculated to lead to the discovery of admissible evidence concerning, SCO's contract claims – which are based on IBM's copying of AIX and Dynix/ptx code into Linux – as well as IBM's recently-added counterclaim – which seeks a broad declaratory judgment that none of IBM's Linux activities infringe any SCO copyright.

IBM has incorrectly resisted producing earlier versions of AIX and Dynix/ptx on the stated ground that SCO need only compare UNIX System V code to Linux code and, therefore, these two sources of code are everything SCO needs to support its claims. But IBM's argument overlooks the pivotal point that the heart of SCO's complaint – the contract claims – requires no proof of any copying from UNIX System V to Linux. Rather, while IBM focuses on copyright law, the license agreements at issue here expressly provided greater protections, ensuring that any subsequent derivatives of UNIX System V derivatives and/or any "methods or concepts" contained in UNIX System V or any such derivatives receive the same protection under the license agreements as the original licensed System V product. Thus, the license agreements, on

their face, clearly provide SCO with far greater protection than the copyright law affords; indeed, the agreements' language would have been surplusage if they had mirrored the copyright protections already provided by operation of law.

IBM's contract argument provides further compelling reasons for SCO's entitlement to design documents, white papers, and programming notes related to the prior versions of AIX and Dynix/ptx. Because IBM's contract argument departs substantially from the language of the operative license agreements, it will necessarily rely on extrinsic evidence. Although SCO believes that the unambiguous meaning of the contracts will preclude IBM's attempt to vary their terms through such extrinsic evidence, to the extent such evidence is permitted, the parties' performance under and understanding of the agreements will undoubtedly be relevant. Thus, SCO is entitled to discover facts showing, for example, that IBM and Sequent understood the restrictions of the contracts to have protected more than distribution of literal UNIX source code, that they gave weight to the language of the contracts, and that they had incentives to breach the contracts.

IBM wants to hold this evidence back, even though (as it elsewhere tells the Court) IBM intends to move for summary judgment on those same contract claims. Apparently, IBM wants the Court to accept its version of the case as a matter of law, yet shield from discovery the very material that could show that at the times that mattered, the licensees themselves did not believe what IBM would now ask the Court to conclude.

IBM also wants to hold back substantial evidence that could show that IBM (and Sequent) breached the contracts in question. It is undisputable that the purpose of those contracts was to allow IBM and Sequent to make and market products based on UNIX System V. The

contracts, granting the licensor more protection than copyright would have done without any agreement, expressly treated the first IBM or Sequent version as subject to the same restrictions that applied to System V itself – as the “original software product.” Any subsequent version, based on that contractually redefined “original software product” would be subject to the very same restrictions – the restrictions that stand at the center of this case, and that would have prevented the ultimate transfer to Linux of the IBM and Sequent programs at issue. IBM wants to shut off access to all of the early development history of AIX and Dynix, even to the early distributed releases of these programs. But this is the very material that would show just how the contractual restrictions applied, something that cannot be shown if access to evidence only begins comparatively late in the development process – as opposed to the point when the contracts and the process began.

IBM’s claim that this material is irrelevant – when this material could contain proof of one instance after another of breach -- is indefensible. IBM is arrogating to itself the power to tell the Court what evidence it should be allowed to see in order to test the validity of IBM’s own versions of the facts.

Wholly apart from SCO’s contract claims, the discovery that SCO seeks is now relevant, and indeed critical, to SCO’s ability to defend against IBM’s recently-added copyright claim. The inordinately time-consuming and labor-intensive task of investigating the entirety of UNIX System V and Linux for evidence of non-literal copying – an issue central to IBM’s counterclaim – is not practical for anyone to perform without leads. The basic discovery that SCO seeks but has not received – such as the identities and precise contributions of AIX and Dynix programmers, which this Court has already ordered IBM to produce – would permit SCO

sensibly to identify the key programmers for deposition. Then, based on such deposition testimony concerning the programmers' access to UNIX source code and the similarity between that code and derivative code in AIX or Dynix, SCO could streamline its investigation by prioritizing and targeting specific areas of code for analysis. By depriving SCO of basic information concerning the contributions of its programmers, IBM seeks to force SCO to undertake this laborious investigation in the most inefficient way possible.

In this and other regards (further detailed below), IBM misapprehends the essence of the discovery process. Discovery is designed to make the investigative process efficient and conducive to truth-finding. IBM has no more right to undermine these goals by blocking discovery than it does to tell the Court what evidence the Court and the jury may see to test the validity of IBM's own claims. Moreover, the question with respect to SCO's requested copyright discovery is not whether SCO possesses some "minimum" amount of evidence that might permit it to prove or defend the claims at issue. Rather, SCO is entitled to the production of all admissible evidence and materials reasonably calculated to lead to the discovery of admissible evidence, unless the burden to IBM outweighs this broad right in a legally and factually significant manner.

In light of the centrality of this evidence to the claims and counterclaims in this case, IBM is hard-pressed to establish any substantial burden that could justify its failure to produce such material. Most of the materials SCO has requested are stored electronically in a centralized, readily accessed system, and, for those that are not, IBM has afforded SCO no opportunity to meet and confer in order to address the issue of producing hard copies. With respect to Dynix/ptx, IBM makes no burden argument of any kind; with respect to AIX, after telling the

Court that it would take “many, many months” to meet the request, IBM now says it would take only “weeks.” Almost every aspect of discovery in this case has taken “weeks” -- over fifty weeks have passed since SCO requested the materials, for example, and over twelve weeks have passed since the Court ordered IBM to produce certain of the documents.

IBM further asserts that all of the paper documents SCO seeks (such as programmer’s notes, design documents and white papers) are too burdensome to produce. IBM makes these statements even though (i) at least a great number of the documents created by programmers are electronically stored, and IBM took the position that SCO would not receive even electronically stored documents, making clear that IBM did not intend to produce any of the discovery to SCO; (ii) when faced with the same issue, SCO agreed to produce documents to IBM in an electronic format; and (iii) by its own admissions, IBM can produce the requested materials in a compact, electronic format. These facts belie any suggestion that IBM is concerned about the burden involved in searching for hard copies, and demonstrate that IBM refuses to produce the discovery at issue in any format.

IBM has repeatedly (and erroneously) argued to this Court and to Judge Kimball that SCO only wants to delay and prolong this litigation in order to further an alleged scheme to create “fear, uncertainty and doubt.” IBM Resp. at 3-4. IBM created this cloud over Linux when it copied AIX and Dynix/ptx code into that program in derogation of its express contractual obligations. And when IBM, to increase its profits and advance its strategic objectives, encouraged numerous entities to migrate to Linux, it imposed the risk it had created for itself on them as well. Here, IBM again tries to shift responsibility, this time claiming that

when SCO seeks basic discovery and IBM stonewalls for a year, the resulting delay is somehow part of SCO's "scheme" to extend the litigation process.

For a year, SCO has sought, and continues to seek, critical discovery to which it is entitled under the Federal Rules in order to present its strongest case to a jury. If IBM has nothing to hide (as it claims), and actually wishes to expedite this litigation (as it also claims), it should produce the materials SCO has requested and attempt to establish its defense (as well as its new copyright counterclaim) on their merits.

ARGUMENT

I. SCO'S DISCOVERY REQUESTS ARE RELEVANT AND REASONABLY CALCULATED TO LEAD TO ADMISSIBLE EVIDENCE TO SUPPORT ITS CONTRACT CLAIMS AND TO DEFEND AGAINST IBM'S COPYRIGHT COUNTERCLAIM

SCO has properly requested all versions of AIX and Dynix/ptx; the names of all principal contributors to AIX and Dynix; and revision information including access to CVMS, all programmer notes, design documents, and white papers. All that IBM has produced are actual commercial releases of AIX and Dynix since 1999 -- releases that IBM characterizes as "distributed." Thus, SCO has not even been given commercially distributed versions prior to 1999. Nor has IBM produced any of the other requested materials, even though most of that material is stored electronically.

Under Federal Rule of Civil Procedure 26, SCO is entitled to this discovery -- which is not only directly relevant, but potentially dispositive of central claims in the case -- unless IBM can show that producing the evidence would constitute such an undue burden that even plainly

relevant material should be withheld. Fed. R. Civ. P. 26(b)(1). IBM does not and cannot make any such showing.

As an initial matter, IBM incorrectly claims in its latest discovery memorandum that it has produced all “distributed” versions of AIX and Dynix since 1999. IBM Resp. at 6, 10-11. But IBM has still indisputably failed to provide at least one distributed version of AIX – version 5.0, see Initial Memo. at 6, n.7; Declaration of Chris Sontag dated July 12, 2004 (“Sontag Discovery Decl.”) ¶ 13 – and IBM offers no explanation for this deficiency. Moreover, SCO’s request for all iterations and versions of AIX and Dynix was not limited to “distributed” products. IBM’s claim that it has complied with SCO’s request because AIX consists only of distributed versions is flatly contradicted by IBM’s own discovery requests in this case, which expressly acknowledge that AIX is not limited to distributed versions, but that AIX is “the UNIX-branded operating system distributed and/or developed by IBM, including all prior versions, releases, and maintenance modifications.” IBM’s First Request for the Production of Documents at 16, ¶ 1 (emphasis added).

A. The Information SCO Seeks Is Discoverable to Support SCO’s Contract Claims

IBM argues that SCO needs only to compare UNIX System V source code with Linux source code to make its case and, therefore, that the discovery SCO seeks is irrelevant. But IBM’s argument ignores the fact that SCO’s principal claims are, as they have been since the beginning of this litigation, for breach of IBM’s contractual obligations under the IBM and Sequent Software License Agreements. Under those contracts, which prohibited IBM from contributing AIX or Dynix/ptx source code into Linux (because such contributions consisted of

derivative works and/or “methods or concepts” of protected works), SCO is not required to show literal or non-literal copying of UNIX System V source code into Linux.¹

Contracts, of course, may be used to secure greater legal protection than that afforded by copyright law alone. See, e.g., Bowers v. Baystate Technologies, Inc., 320 F.3d 1317, 1327 (Fed. Cir. 2003) (affirming jury verdict of breach of contract where “[t]he shrink-wrap license agreement prohibited, *inter alia*, all reverse engineering of [plaintiff’s] software, protection encompassing but more extensive than copyright protection, which prohibits only certain copying”); Dispatch Automation, Inc. v. Richards, 280 F.3d 1116, 1120-21 (7th Cir. 2002) (“contractual alternatives to copyright may give an owner of computer software more protection than copyright would”); Nat’l Car Rental Sys., Inc. v. Computer Assocs. Int’l, Inc., 991 F.2d 426, 431-35 (8th Cir. 1993) (finding that party alleging infringement was in actuality asserting that the infringer violated a licensing agreement by processing data for third parties, which was not an infringement of an exclusive right to the copyright holder and that “Absent the parties’ agreement, this restriction would not exist”).

The IBM and Sequent License Agreements secured such additional protections for SCO; if they had not, the express protective language of the agreements would have had no meaning, constituting inexplicable surplusage. Consequently, SCO can prevail on its contract claims without proving IBM’s liability for copyright infringement.

IBM therefore misses the point when it claims that prior versions of AIX and Dynix are irrelevant on the ground that “[a] derivative work based on an original work may be so transformed that it no longer is a derivative work and no longer infringes any copyright of the

¹ For the reasons explained in Part I.B, below, IBM’s argument would be wrong even if this case were based entirely on a Linux-related copyright claim like the one IBM added in its Tenth Counterclaim (but SCO has never asserted).

original work.” IBM Response at 14 & n.5. IBM’s argument assumes that only copyright law applies. But SCO’s Linux-related claims in this case are contract claims, and the contracts at issue were expressly worded to give SCO the protection that IBM claims copyright law would not.

This is also precisely the reason why earlier versions of AIX and Dynix are so critical to SCO’s contract claim proof. Specifically, under Paragraph 2.01 of the IBM and Sequent license agreements, “modifications and derivative works based on” UNIX System V are treated as if they were part of the original licensed “SOFTWARE PRODUCT,” so that any transfer by IBM of such a modification or derivative work – even if it was included in an early version of AIX and Dynix – would violate the License Agreements. Similarly, under the Sequent License Agreement, IBM is specifically prohibited from disclosing “any or all of such SOFTWARE PRODUCTS [i.e., UNIX System V and, pursuant to Paragraph 2.01, AIX and Dynix] (including methods or concepts utilized therein) to anyone,” so that any disclosure by IBM of any portion, method, or concept of Dynix – even if was included in an early version of the program – would violate the Sequent agreement.

SCO’s need for this discovery is especially critical in light of IBM’s stated intention to seek summary judgment on SCO’s contract claims. See IBM SJ Mem. at 5 n.4. If and when IBM brings such a motion, it will argue (as it must) that SCO has not produced sufficient evidence to raise any issues of material fact regarding its contract claims. In a similar vein, IBM has already moved for summary judgment on its own recently-filed Tenth Counterclaim, arguing that SCO cannot demonstrate any issues of material fact regarding copyright infringement by

IBM in any of its (vast and continuously expanding, but only partially enumerated) “Linux activities.”

IBM’s litigation two-step has thus become clear. First, IBM systematically withholds relevant discovery materials in the face of SCO’s requests and this Court’s Orders. Then it moves for summary judgment, arguing that SCO cannot produce evidence sufficient to raise a fact issue on its claims. IBM’s method of defense litigation runs contrary to both the letter and spirit of the Federal Rules of Civil Procedure and this Court should order IBM to comply with its discovery obligations.

1. The Requested Materials May Contain IBM Admissions of Contract Liability.

In its Initial Memorandum, SCO requested not only intermediate versions and iterations of AIX and Dynix/ptx, but also revision history information about these programs, as well as programmers’ notes, design documents, and white papers. These materials will provide SCO with extensive information about IBM’s programmers and their involvement with AIX and Dynix/ptx. Declaration of Chris Sontag submitted July 8, 2004 (“Sontag SJ Decl.”) ¶¶ 51-53. The materials SCO seeks may also contain evidence that is not only probative, but dispositive of IBM’s liability for breach of contract, such as:

- admissions concerning the meaning of, and limitations imposed by, the license agreements;
- admissions regarding IBM’s liability for breaching those contracts; and
- admissions that the development of AIX and Dynix/ptx depended on UNIX System V.

It is difficult to understand IBM's claim that such materials are not relevant to this case -- when they could easily be dispositive of this case. It is even harder to understand how IBM could continue to try to withhold this material, when it has told the Court that it plans to move for summary judgment on the contract issues, and when all of these materials could readily contradict the very claims that IBM will be asking the Court to then accept as undisputed.

2. SCO's Requested Discovery Is Directly Relevant to Whether IBM's Disclosure of AIX and Dynix Violated the Software License Agreements.

The License Agreements in question allowed IBM and Sequent to use System V source code to prepare operating systems (AIX and Dynix/ptx, respectively) based on System V, subject to several restrictions. Critically, Paragraph 2.01 of the License Agreements provides that any modifications or derivative works based on System V code are to be treated as "part of" System V (and thus subject to the restrictions applicable to System V) under the Agreements. Because of this clause in the Agreements, IBM is no more free to disclose, export, or publish any derivative version of AIX or Dynix/ptx than it is to disclose, export, or publish System V itself.

Accordingly, one way for SCO to prove that IBM breached the License Agreements is for SCO to demonstrate that versions of AIX and/or Dynix/ptx are derivatives of System V within the meaning of the Agreements, and that IBM disclosed, exported, or published those derivatives. As the plain language of the License Agreements makes clear, a modification or derivative of System V is subject to the same confidentiality restrictions as System V itself. Thus, the restrictions in the Agreement apply to each successive iteration of the derivative program and not to System V alone.

To illustrate, suppose that Dynix version 1 is a derivative of System V. Under Paragraph 2.01, every provision in the License Agreement that applies to System V also applies to Dynix v.1 (including Paragraph 2.01 itself). Thus, if Dynix v.2 is a derivative of Dynix v.1, it too must be treated as part of System V for the purposes of the Agreements. Assuming that each version of Dynix is created as a derivative of the previous version, the most recent version of Dynix is every bit as much a derivative of System V under the License Agreements as is Dynix v.1. Accordingly, SCO's request for each iteration and version of AIX and Dynix is discoverable under its breach of contract claims; if SCO can trace the evolution of AIX and Dynix back to System V it will be able to prevail on those claims without making any additional showing.

But there is no way for SCO to make this showing without the early history of AIX and Dynix – the very material that IBM is dead set against providing.

IBM, of course, disputes SCO's interpretation of the contracts and argues that, at some (as yet undefined) point, AIX and Dynix ceased being derivatives of System V. The analysis in Dispatch Automation, 280 F.3d 1116, demonstrates that the requested materials are discoverable even under IBM's theory of the License Agreements. In that case, Judge Posner explained the inherent difficulty in determining the point at which a computer program ceases to be derived from an old program, and instead becomes so transformed that it is a new work:

To decide . . . whether a successive version of a computer software product is a merely incremental improvement or a breakthrough would be like determining the exact moment at which day becomes night. . . . How for example to decide whether to compare the latest version of a product with the immediately preceding version or with the original version? Compared to the former, it might seem incremental, while compared to the original version it might seem discontinuous, a radical or fundamental change. The first approach would invite the proliferation of meaningless intermediate steps designed to disguise novelty; but the second would ignore the obvious fact that a process of unmistakable, indeed quite

gradual, development can have an end point radically different from its beginning, as in the case of the oak and the acorn it grew from.

Id. at 1119-20.

By resisting SCO's discovery, IBM would arrogate to itself the authority to select evidence with which to test the validity of its claim that at some point AIX and Dynix/ptx ceased to be modifications or derivative works "based upon" SCO's UNIX System V and were no longer subject to the restrictions of the License Agreements. In other words, IBM itself would select that point, without ever allowing the Court, SCO, or the jury to see any evidence that IBM did not want it to see – any evidence of the intermediate stages of development. IBM would also decide how to deny SCO the discovery it needs to prove where that actual point should be located.

If this approach to discovery were permissible, there would be no reason for licensors ever to enter carefully worded license agreements, because they would be literally worthless. Another factor makes IBM's approach even less appropriate: SCO's License Agreements are worded precisely to avoid the complexity of this very evolutionary determination. That is, SCO's contracts with IBM and Sequent make every modification and derivation part of the original "SOFTWARE PRODUCT" and subject to all of the contract's prohibitions and limitations. Following Judge Posner's analogy, accepting IBM's position would be attempting to determine "the exact moment at which day becomes night" with all the blinds drawn. Id. That is the exact type of determination the licensing agreements were drafted to avoid.

3. The Requested Discovery is Relevant to and Will Aid SCO in Determining Whether IBM Disclosed Contractually-Protected System V "Methods or Concepts."

Paragraph 7.06 of the Sequent License Agreement prohibits IBM from disclosing any portion (let alone the entirety) of any "SOFTWARE PRODUCT" (*i.e.*, original System V and derivative (Dynix/ptx)), "including methods or concepts utilized therein." Therefore, SCO will prevail on its breach of contract claims if it can demonstrate that IBM in fact disclosed protected "methods or concepts." The most efficient and straightforward way for SCO to discover if IBM made such disclosures is through access to Dynix/ptx programmers and their particular contributions.

In its March Order, this Court directed IBM to produce, among other things, the identities and precise contributions of Dynix programmers. See 3/3/04 Order at 5. SCO has now attempted to obtain this information from IBM in several ways: by serving an interrogatory, filing a motion to compel, and filing a renewed Motion to Compel (on July 6, 2004). Up to this point, IBM has resisted these efforts and withheld this essential discovery. Accordingly, SCO requests that IBM again be ordered to produce materials that will reveal the identities and precise contributions of Dynix programmers.

In the first instance, this information would allow SCO to take depositions of the principal Dynix contributors. These depositions could result in admissions of IBM's reliance on protected SCO material, clearly relevant to IBM's potential contract liability. Programmers and engineers can be deposed regarding the following issues:

- the identify of Linux contributors;
- specifics about their own Linux contributions;
- assistance given to Linux contributors; and
- whether any protected "methods or concepts" were ever disclosed.

This type of testimony will also assist SCO in identifying other former IBM employees contributing to Linux, who may be valuable witnesses.²

Accordingly, SCO's requested discovery is clearly relevant to, and reasonably calculated to lead to the discovery of admissible evidence concerning, SCO's contract claims in this case. In fact, the requested material may well contain much that is not only probative – but dispositive – of SCO's contract claims. IBM's argument that such material is not relevant reduces, again, to a demand that IBM be allowed to tell the Court what evidence should be allowed to be used to test the validity of IBM's own claims.

B. SCO's Requested Information Is Discoverable to Defend Against IBM's Recently-Added Copyright Claims.

SCO's case against IBM consists primarily of breach of contract claims. SCO's only copyright claim arises out of IBM's breach of the IBM and Sequent licensing agreements.³ Nowhere in SCO's complaint is there a Linux-based copyright claim.

On March 29, 2004, however, IBM injected Linux copyright issues into this case through its Tenth Counterclaim, which seeks a broad declaration that none of its Linux activities infringe any of SCO's copyrights. Less than two months later, IBM filed its motion for summary judgment wherein it asserts that SCO must be limited to facts contained in SCO's discovery responses from January 12 and April 19 of this year. IBM SJ Mem. at 7, 23, 28, 30. Thus, having recently introduced new Linux-related copyright claims into this case, IBM is seeking to prevent SCO from conducting any discovery on IBM's new Tenth Counterclaim.

² IBM claims that programmer information is contained in the AIX and Dynix programs themselves and that, consequently, IBM need not provide such information independent of AIX code. SCO shows why IBM's claim is wrong in its renewed Motion to Compel (July 6, 2004) regarding its Interrogatory No. 5.

³ SCO's copyright claims are based on claims that IBM transferred UNIX code to India, a country where the licensing agreement prohibited its use, and that IBM continued to use and distribute UNIX after SCO had terminated the licensing agreement.

While IBM may prefer such an arrangement, its interests do not dictate the rules of discovery. SCO's discovery requests at issue in this memorandum, though initially made to support its contract claims, are now clearly relevant and necessary to SCO's defense against IBM's copyright claim.

The principal way in which IBM could defeat IBM's declaratory judgment counterclaim is by showing that IBM has copied literal code – or elements of non-literal expression – from SCO's copyrighted material. In refusing to comply with SCO's discovery requests, IBM tries to limit SCO to the least efficient and most time-consuming of all possible manners of investigating IBM's copyright counterclaim. IBM's position is indefensible as a matter of law, would be unjustifiable even if IBM had not injected the copyright claim into the case only recently and then even more recently moved for summary judgment on that claim, and is particularly inexplicable in light of IBM's constant complaints about delay. Indeed, the discovery SCO requested over a year ago would allow it to streamline, narrow, and prioritize its discovery analysis, and thereby speed up the litigation process.

While there are a number of ways to show that IBM copied SCO's protected material, perhaps the most obvious is for SCO to demonstrate that protected material, whether UNIX System V code – or non-literal expression – is present in Linux. IBM would then be liable as an end-user of Linux and possibly as a contributor to Linux as well. To do this, SCO must undertake a complex and laborious comparison of Linux and UNIX System V. Sontag SJ Decl. ¶¶ 9-14. This is a monumental task. Both the UNIX and Linux operating systems are large and complex computer programs with many millions of lines of code. In addition, SCO needs to do more than merely examine the two programs for literal, line-for-line character similarity, a

daunting process in its own right. That literal comparison must be supplemented by an even more time-consuming second order review. Even IBM does not (because it cannot) contest that copyright law protects more than literal line-for-line copying -- that it also protects copying of structures and sequences, even absent any copying of literal code. Review for such structure and sequence copying is even more time-consuming and laborious, if forced to be done without the aid of discovery, which is what IBM is trying to accomplish here.

Initially, each of the millions of lines of Linux code must be compared with each of the millions of lines of UNIX System V code, line-by-line, or in groups of lines according to the structure, sequence, or function of the code. SCO can use an automated tool to perform the initial literal comparison, but automated tools do not work well in the absence of identical copying. Sontag SJ Decl. ¶¶ 10-13. Those tools identify only exact or nearly-exact literal matches, thereby excluding evidence of non-literal copying.

These inherent limitations of automated code comparison tools mean that most code comparison between Linux and UNIX will require a manual review by a skilled programmer or expert, an extremely time-consuming process. The time required to complete the code comparison without some shortcut -- such as a roadmap or list of "hot spots" in Linux -- is staggering, on the order of thousands of man-years.⁴ SCO needs to use discovery to streamline,

⁴ To execute the code comparison, SCO and its experts must compare page after page of code. The 4 million lines of Linux kernel code, which is the core portion of UNIX, takes up 66,000 pages; the UNIX kernel comprises 3.4 million lines of code and takes up 58,000 pages. A simple manual comparison would involve placing the pages of code side by side in some ordered manner and then looking for the same or similar structure, sequence, and organization of the code. Assuming each page comparison takes one minute, and that there are 66,000 x 58,000 comparisons, this "initial" review could take on the order of 25,000 man-years. Following the initial review, SCO and its experts must conduct a detailed comparison of likely copying candidates. This "second-level" review would also require extraordinary human labor and consume enormous time. See Sontag Decl. ¶ 14.

narrow, and prioritize this manual review in order to avoid what could otherwise be an unrealistically difficult and time-consuming effort.

Many shortcuts to completion of the code comparison are theoretically available, including: (1) reviewing directory and file structures; (2) searching for similar file names; and (3) reviewing the AIX and Dynix/ptx development history, including reviewing interim and published versions of AIX and Dynix/ptx, as well as the associated documents, such as the version control systems (e.g., IBM's CMVC system), bug tracking logs, white papers, design documents, and programming notes. Sontag SJ Decl. ¶¶ 26-35, 51-53.

SCO has already investigated the viability of the first two shortcuts and, even though those tools are helpful, the breadth of the analysis remains daunting.⁵ The third shortcut, however, can greatly streamline SCO's further discovery. Specifically, by reviewing the development of AIX and Dynix/ptx, and their associated documents, SCO would be able to: (1) track original UNIX System V code into Linux and thereby focus its efforts by matching up original UNIX code with its Linux counterpart and (2) identify IBM and Sequent programmers and engineers who can provide deposition testimony that would also allow SCO to focus and prioritize its efforts to locate Linux code substantially similar to UNIX System V code. Sontag SJ Decl. ¶¶ 15-25.

As noted in the accompanying Declaration of Christopher Sontag, IBM's version control system (VCS) – the Configuration Management/Version Control (CMVC) system – can help

⁵ One shortcut to comparing the UNIX and Linux code might be comparing similar directory structures of the UNIX and Linux operating systems. For example, version 2.4 of the Linux kernel contains 530 subdirectories and about 8750 source and assembly files. See Understanding The Linux Kernel, Bovet. Assuming each of the 8750 files requires on average one day to investigate (some files are small, but others contain thousands of lines of code), about thirty-five man-years would be required to review all the Linux kernel files in detail. However, this calculation ignores the possibility that the two operating systems use different file names and a different file layout, and that similar code sequences may reside in entirely different files. See Sontag Decl. ¶ 15.

SCO identify the programmers and engineers familiar with relevant UNIX-based code that IBM or third parties contributed to make Linux enterprise-hardened and multiprocessor-capable. Sontag Discovery Decl. ¶¶ 4-5, 7. SCO can then take their depositions and use that information to focus its analysis of UNIX System V and Linux.

The VCS will also allow SCO to examine the development of AIX and Dynix/ptx. By examining the source code in successive versions of AIX and Dynix, SCO can trace its UNIX System V code through to current versions of AIX and Dynix and determine where in Linux SCO's UNIX System V code is copied.⁶ Then, SCO will know exactly which portions of UNIX System V and Linux to group together for comparison. This will significantly streamline SCO's efforts to find code in Linux that is substantially similar to UNIX System V code.⁷

IBM contends that because SCO has UNIX System V code and access to Linux code, SCO does not require interim versions or revision history information and, therefore, needs nothing further to make its copyright infringement case. IBM Resp. at 13-14.⁸ But IBM

⁶ Software (i.e., source code) undergoes many changes during its initial development and later over its operational life. Changes may occur as frequently as daily and can continue for years. Software changes typically are driven by the need to correct "bugs," to improve features, or to add new features. Because of changes made to source code over time, a current code version may "look" different than the initial code version, making identification of the initial code version difficult, and substantial similarity and derivation more difficult to establish.

⁷ The Sontag Declaration describes how such an examination of code lineage would streamline SCO's discovery efforts. Mr. Sontag provides an example of SCO's UNIX System V source code for a print error function (perror.c), illustrating accumulated modifications over time. The first version of the code appeared in 1981, and by the time the seventeenth version appeared more than ten years later, the perror.c code sequence was almost unrecognizable from the initial version. Yet a careful review of the code reveals that the seventeenth and the initial versions have the same structure, sequence, and organization – the accepted test to show substantial similarity. See Gates, 9 F.3d at 836-839. Table 1, which shows the incremental change to perror.c code over time by way of code difference plots, makes explicit what SCO contends here: access to IBM's CMVC system would greatly streamline SCO's discovery efforts. See Sontag Decl. ¶¶ 37-42.

⁸ In support of its argument that SCO needs no further discovery, IBM relies on the plainly inapposite case of Gemisys Corp. v. Phoenix Am., Inc., 186 F.R.D. 551 (N.D. Cal. 1999). IBM Resp. at 14. In Gemisys, the plaintiff sought to depose three witnesses for a second time in the apparent hope that they would change their testimony, and to depose an undisclosed number of other witnesses on issues about which it had already obtained information

fundamentally misconstrues its discovery obligations. SCO is entitled to the production of all materials relevant or reasonably calculated to lead to the discovery of admissible evidence, unless IBM can show some undue burden. Fed. R. Civ. P. 26(b)(1).

Indeed, IBM cites no authority requiring that SCO be limited to only one method in undertaking to prove that UNIX System V and Linux are substantially similar. That limitation would be the outcome of the decision IBM seeks. SCO would be precluded from knowing the precise contributions made to AIX and Dynix and would therefore be precluded from deposing the significant authors of AIX and Dynix. It would thus be unable to streamline, narrow, and prioritize its searches for code and non-literal elements in Linux that originated in UNIX. Without this discovery, the investigation is much more time consuming, costly, and needlessly inefficient.

The rules of discovery are designed to increase efficiency and to enhance the likelihood that the truth will be brought out. The rules are not, by contrast, designed to give one party the prerogative to make the investigative process as inefficient as conceivably possible and to enhance the likelihood that if the truth ever comes out, it will be long after the claims against it have been adjudicated. IBM has no more right to use the discovery rules to this end than IBM does to tell the Court what evidence the Court and the jury may see to test IBM's own claims.

II. IBM FACES NO SIGNIFICANT OR UNDUE BURDEN IN PRODUCING THE CRUCIAL DISCOVERY SCO HAS SOUGHT FOR OVER A YEAR.

through prior depositions. *Id.* at 565-66. In a Rule 56(f) motion, the plaintiff requested this redundant discovery based on its mere "hope that further evidence will develop prior to trial." *Id.* at 566. In contrast, SCO's request is not based on a speculative hope that evidence will develop, but on SCO's specific demonstration of the relevance of the discovery material to SCO's case and the lack of burden IBM will bear in producing it. Moreover, SCO requests discovery thus far withheld by IBM in contravention of SCO's requests and this Court's Orders, and does so through a motion to compel, where "the concept of relevance should be construed very broadly." 3/3/04 Order at 3 (quoting *Gohler, IRA et al. v. Wood et al.*, 162 F.R.D. 691, 695 (D. Utah 1995)).

The party opposing discovery carries the burden of showing that “the burden or expense of the proposed discovery outweighs its likely benefit” under Fed. R. Civ. P. 26(b)(iii). See, e.g., Simpson v. Univ. of Colo., 220 F.R.D. 354, 359 (D. Colo. 2004) (holding that “when the discovery sought appears relevant,” the party resisting disclosure on the ground of burdensomeness must demonstrate that its purported burden “would outweigh the ordinary presumption in favor of broad disclosure”) (internal quotation marks and citations omitted).⁹ The party opposing discovery must demonstrate some “plainly adequate” reason not to produce relevant material. 8 C. Wright et al., Federal Practice and Procedure § 2035 (2003). With respect to electronically available information, “whether production of documents is unduly burdensome or expensive turns primarily on whether it is kept in an *accessible* or *inaccessible* format.” Zubulake v. UBS Warburg LLC, 217 F.R.D. 309, 318 (S.D.N.Y. 2003).

IBM fails to show that the requested discovery is inaccessible, or that its burden in producing the discovery would be significant or would outweigh the crucial information (as outlined above) that SCO seeks through the discovery.

IBM Claims No Burden as to Dynix/ptx. While IBM contends that producing the requested information about AIX that is electronically stored would take “many weeks,” IBM makes no claim that there would be any similar burden involved in doing so for Dynix. With respect to AIX, IBM’s principal argument is that the CMVC on which AIX source code is stored does not permit IBM easily to copy that code for production to SCO -- thus the requirement of “weeks” of work. IBM Resp. at 18-19. At the same time, IBM makes no argument that the version control system on which the Dynix source code is stored, Revision Control System

⁹ Magistrate Judge Boyce often remarked that a showing of undue burden sufficient to defeat a request for relevant discovery must be akin to “cleaning the Aegean stables.”

(RCS), precludes IBM from easily copying that code for production to SCO. IBM's supporting affidavit, required to support a claim of burden, see, e.g., Super Film of Am., Inc. v. UCB Films, Inc., 219 F.R.D. 649, 657 (D. Kan. 2004), makes no mention of RCS and Dynix. In the absence of any argument on the issue, IBM necessarily fails to meet its burden with respect to Dynix, including all of the electronically stored programmer notes and design documents and other requested documents, and the identities and precise contributions of programmer contributors to Dynix (which are also the subject of a renewed motion to compel before this Court).

IBM's Claim Cannot Avoid Producing AIX Versions and Related Electronically Stored Documents By Contending That It Would Take "Weeks" to Do So. IBM previously told this Court that it would take "many, many months" for IBM to produce the information, 2/6/04 Tr. at 37:10-14, which IBM now claims would take "many weeks" for IBM to produce (but presumably not a full month). Even assuming IBM is now correct, this burden of "weeks" is not a remotely adequate basis under the law for allowing IBM to withhold the requested materials. Almost every aspect of discovery in this case has taken "weeks" -- over fifty weeks have passed since SCO requested the materials, over twelve weeks have passed since the Court ordered IBM to produce certain of the documents, and the deadline for fact discovery is eight months away. On these bases alone, under the circumstances, IBM does not face a significant burden. The materials are a likely source of probative -- and dispositive -- evidence with respect to SCO's contract claims; the material in fact stands at the center of the case. The material also is essential to streamlining an otherwise unnecessarily time-consuming investigation into non-literal copyright violations, on claims IBM has recently introduced into the case. The proposal by IBM that it should be able to withhold such material from the Magistrate Judge, the Court, and the

jury because producing it would inconvenience IBM by a matter of “weeks” is simply not a serious argument.

IBM’s Claim Regarding the “Number” of Documents Is Inapposite. IBM claims that the information sought by SCO “amounts to millions of pages of documents,” created by programmers and “40 million additional pages of paper” for additional source code. IBM Resp. at 3. IBM makes these statements notwithstanding that (i) at least a great number of the documents created by programmers are electronically stored, and IBM took the position that SCO would not receive even electronically stored documents, making clear that IBM did not intend to produce any of the discovery to SCO; (ii) when faced with the same issue, SCO agreed to produce documents to IBM in an electronic format; and (iii) by its own admissions, IBM can produce the requested materials in a compact, electronic format. Thomas Decl. ¶ 11. These facts belie any suggestion that IBM is concerned about the burden involved in searching for hard copies, and demonstrate that IBM refuses to produce the discovery at issue in any format.

The foregoing points -- without more -- are independently sufficient to preclude IBM’s claims of “burden.” It makes no claim with respect to Dynix. Its claims on AIX are trivial, measured against the importance of the evidence, the other discovery tasks in the case, and simply against the kinds of efforts IBM has forced SCO to pursue just to try to obtain rudimentary discovery requested at the very outset of this case (burdens that have required SCO to renew motions to compel and spend much more than “weeks” of effort). But as shown below there are even further problems with IBM’s claims.

IBM’s Claim of “Weeks” to Produce Certain Discovery Is Inaccurate. To the extent the period of “many weeks” is even significant, the question as to AIX is whether it would actually

take IBM, one of the world's largest computer companies, that much time to produce earlier versions of the AIX source code. One reason to doubt IBM's representation is that, as noted above, IBM originally told the Court that it would take "many, many months" for IBM to produce the information. 2/6/04 Tr. at 37:10-14. Indeed, IBM's prior production in this case illustrates how long it would take IBM to copy AIX from data tapes onto DVDs or CDs: when IBM produced versions of AIX in unreadable format on March 4, 2004, SCO sought and received replacement CDs three weeks later.¹⁰

IBM's own pre-litigation statements further belie its current assertion that CMVC faces significant limitations that would make the production of AIX source code difficult:

- IBM promotes CMVC to its customers based on CMVC's flexibility and sophistication, and IBM touts the use of automated scripts to incorporate fixes and changes daily. IBM-Siebel Marketing at 23 (Sontag Decl. Exh. 5).
- IBM states that "CMVC is broad in its application, thorough in its implementation, and very flexible." IBM's Redbook, Did You Say CMVC?, Sept. 1994, at 65 (Sontag Decl. Exh. 2).
- According to IBM, "The most important thing to understand about CMVC is that you do not need to understand all of it, before you begin using some of it. . . . CMVC provides many mechanisms to help you accomplish your SCM [software configuration management] goals, but it does not dictate exactly how you should use them, nor does it require that you use them all if you use only some of them." Id.
- IBM also claims that "CMVC ensures that an audit trail is maintained for every file by identifying for any file change: when the change occurred, who was responsible, and why the file was modified." Id. at 9.
- One IBM employee explained that the "VC" (Version Control) function of CMVC "eliminates confusion when looking for a particular source code file belonging to a particular product," which describes what SCO seeks to do here. Email from Adrian

¹⁰ Given that IBM initially defended its decision to produce the code on DDS tapes and refused to provide them in the DVD format, it obviously took less than three weeks to make the necessary transfer once they had decided to provide them to SCO in the correct format.

Mitu of IBM Canada, Aug. 19, 1993 (located at <<http://www.cs.queensu.ca/Software-Engineering/blurb/cmvc>>) (Sontag Decl. Exh. 3).

- The same employee indicates that CMVC data may be accessed remotely, implying that IBM could simply grant SCO remote access to CMVC data to respond to SCO's discovery requests. Id.
- IBM also acknowledges the possibility of granting users remote access to CMVC data, as long as the user has CMVC software. IBM's Redbook at 209-10.

Other, confidential IBM documentation further explains why it would not be burdensome for IBM to access source code and documentation from CMVC. See Sontag Discovery Decl. ¶¶ 5-7, 11-12.¹¹

If the foregoing leaves any room for doubt, SCO demonstrates in the accompanying declaration of Chris Sontag that IBM grossly exaggerates the burden it will have to undertake to provide CMVC material. IBM can easily provide its CMVC system, including AIX source code and associated documentation, Thomas Decl. ¶ 7, in electronic format -- IBM admits that the total file size is manageable, on the order of 40 gigabytes, id. ¶ 10. This amount of materials could easily fit onto a laptop computer's hard drive, a portable hard drive, or a few optical disks. CMVC is arranged using IBM's SCCS hierarchy, see IBM's Redbook at 40, which permits IBM easily to design and run searches (accomplished by short computer programs called "scripts") to identify AIX source code. See Sontag Discovery Decl. ¶¶ 18-26. In the alternative, IBM can use standard archival tools to determine the locations of the one or more SCCS directory hierarchies related to AIX, and make copies. See Sontag Discovery Decl. ¶ 27. Contrary to the assertion in

¹¹ IBM further states that it "does not maintain revision control information for AIX source code prior to 1991" Thomas Decl. ¶ 5 (emphasis added), but does not state whether they possess pre-1991 revision control information for AIX source code. Even if pre-1991 AIX revision control information is unavailable, that does not mean IBM cannot produce pre-1991 notes, white papers and design documents, as well as releases and versions of AIX.

Mr. Thomas's declaration, Thomas Decl. ¶ 9, IBM engineers need not conduct a manual search for the components that are part of AIX, Sontag Discovery Decl. ¶ 29.

IBM also takes pains to describe each step it would take to manually produce the responsive materials from CMVC (IBM Resp. at 9-10), but such steps are unnecessary to accomplish the end result electronically. See Sontag Discovery Decl. ¶¶ 18-28. IBM further claims that to copy AIX information from data tapes onto DVDs or CDs would take "many additional weeks", after the "many weeks" already taken to collect the information. IBM Resp. at 10; Thomas Decl. ¶ 11. In fact, the data extraction and transfer process should not take more than two or three days, if IBM follows standard industry practices of maintaining and downloading its CMVC data. See Sontag Discovery Decl. ¶ 34.

IBM's Claim that SCO Merely Seeks "Delay" Is Baseless. IBM finally contends that SCO's requests are designed to "lay the groundwork for further delay." (IBM Resp. at 19.) The claim is remarkable to say the least, give that SCO here seeks the most basic types of discovery, which it has sought for over a year and has recently filed a Renewed Motion to Compel to ask the Court to order (again) IBM to produce.

The assertion also ignores IBM's (plain) responsibility for the delay in the discovery at issue here. IBM could have participated in a meet and confer process to eliminate or dramatically reduce the perceived burden it believed it faced. Further, IBM declined to meet SCO at least partway on SCO's requests for programmers' notes, design documents and white papers, making a blanket claim that all such documents were too burdensome to produce. IBM's conduct has been inconsistent with their stated intent to avoid "delay."

As set forth in SCO's opposition to IBM's motion for summary judgment on its Tenth Counterclaim, a pattern to IBM's conduct has emerged. IBM repeatedly tells the Court (and the world) that SCO has no case and that SCO seeks only to delay. But a defendant who truly believes its adversary has no case, who truly wishes to avoid delay, does not:

- take every possible step – even in the face of court orders – to withhold access to the information needed to test its position, thereby ensuring great delay; and
- then ask the Court to cut off all work so that its position can never be tested, thereby ensuring that even a plaintiff with a very strong case could never prosecute it.

Discovery battles are common and, to some extent, unavoidable in complex civil litigation. This pattern is not.

SCO submits that for the foregoing reasons, and as also set forth SCO's Renewed Motion to Compel dated July 7, 2004, IBM faces no significant or undue burden in producing the discovery SCO requests.

CONCLUSION

SCO respectfully requests, for the reasons set forth above, that the Court order IBM to produce the materials set forth above and in SCO's opening memorandum.

DATED this 12th day of July, 2004.



HATCH, JAMES & DODGE, P.C.
Brent O. Hatch

BOIES, SCHILLER & FLEXNER LLP
Robert Silver
Stephen N. Zack
Mark J. Heise

Counsel for Plaintiff/Counterclaim defendant

ANDREWS KURTH LLP
Frederick S. Frei
Aldo Noto
John K. Harrop

Of Counsel

CERTIFICATE OF SERVICE

Plaintiff, The SCO Group, hereby certifies that a true and correct copy of its **REPLY MEMORANDUM REGARDING DISCOVERY** was served by United States Mail, first class postage prepaid, this 17TH day of July, 2004, on the following:

Alan L. Sullivan, Esq.
Todd M. Shaughnessy, Esq.
Snell & Wilmer L.L.P.
15 West South Temple, Ste. 1200
Salt Lake City, Utah 84101-1004

Evan R. Chesler, Esq.
David Marriott, Esq.
Cravath, Swaine & Moore LLP
825 Eighth Avenue
New York, NY 10019

Donald J. Rosenberg, Esq.
1133 Westchester Avenue
White Plains, New York 10604


