EXHIBIT 1

[redacted]

IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
TEXARKANA DIVISION

MAXELL, LTD., Plaintiff v. APPLE INC., Defendant.	Civil Action NO. 5:19-cv-00036-RWS **JURY TRIAL DEMANDED** [redacted]

APPLE INC.'S SUR-SUR-REPLY IN SUPPORT OF ITS RENEWED MOTION TO COMPEL INFRINGEMENT CONTENTIONS COMPLIANT WITH PATENT RULE 3-1(G) AND FOR SCHEDULE EXTENSION OR, IN THE ALTERNATIVE, TO PRECLUDE MAXELL'S RELIANCE ON SOURCE CODE FOR INFRINGEMENT

P.R. 3-1(g) and the Court's orders plainly explain what Maxell was required to disclose to Apple with respect to source code Apple produced before February 12. Maxell fell far short of complying with either, and its strategy is plain: force Apple to disclose its expert opinions without the fair notice of the full complement of Maxell's infringement contentions (P.R. 3-1(a) and 3-1(g)) to which Apple is entitled.

Maxell does not and cannot dispute the core facts underlying Apple's motion: (1) Maxell's SSIC lists whole source code files for each claim element (or for groups of claim elements), (2) each file can comprise dozens or hundreds of printed pages and a multitude of software functions, (3) a single software function can typically fit on a single page, and (4) Maxell can—but refuses to—specifically identify which software functions it actually accuses of infringement. Maxell's sur-reply repeats the same illogical argument from its Opposition—that because Apple's "engineers understood and answered questions about the cited source code files," they must have "underst[ood] what accused functionalities" are at issue in this case. D.I. 313 at 1. If this were the test, then P.R. 3-1(g) would be rendered meaningless and the Patent Rules would be turned on their head: it would be a defendant's obligation to dig through thousands of lines of source code across hundreds of files just to divine the single (or few) pages in each file that relates to a plaintiff's infringement theory. But, as the Court already found, P.R. 3-1(g) requires Maxell to give Apple fair notice of its source code contentions and to do so in a manner "sufficiently focused to the accused functionality."

Maxell's Sur-Reply rehashes its argument (already rejected by the Court) that its P.R. 3-1(a) non-source code claim charts, the complaint, *Markman* arguments, and Apple engineers' testimony give Apple sufficient notice and justify its failure to comply with P.R. 3-1(g). Maxell also again touts familiar excuses, like Apple's supposed "discovery delays."[1] But it is Maxell's refusal to provide compliant contentions that necessitates delay (or striking them in part).

I. Maxell Cannot Dispute That It Cites Large Ranges of Code Without Explanation

'493 Patent. Maxell does not dispute that its SSIC cite a dozen source code files for claim element 5.e [redacted] or that hundreds of pages (and functions) are found within those files. D.I. 284 at 2-3; D.I. 306 at 1-2. And Maxell abandons the "textual disclosures" touted in its Opposition, (D.I. 306 at 1-2), and now argues that the *Markman* hearing demonstrates Apple's understanding of this claim element and no further explanation is necessary. D.I. 313 at 2. There is no exception in P.R. 3-1(g) for claim elements construed by the Court. The Court has already rejected Maxell's reliance on screenshots and unsupported conjecture about Apple's software as a substitute for compliance with P.R. 3-1(g). D.I. 156 at 1-2; *see* D.I. 204 at 2. Those contentions do not sufficiently disclose what *source code* functionality Maxell accuses for its designated "software limitations"—that is what P.R. 3-1(g) is for. D.I. 306 at 2.[2] And Maxell's claim that it does not know what more it can do is betrayed by its offer to providing more detailed contentions for the [redacted] element, and its ability to identify and print specific portions of source code. D.I. 313 at 2, 4; D.I. 306 at 4.

'794 Patent. Maxell does not dispute that its SSIC cite 16 files for element 1.f and that just one of those files spans ~200 pages and over 100 functions. D.I. 284 at 3; D.I. 306 at 2-3. Nothing in Maxell's complaint or generic description of source code directory names—each of

[1] Maxell does not dispute that its complaints about Apple's discovery were resolved "largely in *Apple's* favor." D.I. 313 at 1 n.2. That Apple voluntarily produced some documents after Maxell's motion simply confirms Maxell's failure to meet and confer before filing its motion.

[2] Maxell suggests that Apple should simply move to strike Maxell's expert reports (which Apple may be forced to do). D.I. 313 at 2 n.5. But Maxell's failure to crystalize its theories before expert reports is inherently prejudicial to the Court and Apple and severely hinders the Court's and Apple's ability to police those reports. D.I. 284 at 5 (citing *Finjan*).

which contains thousands of files—explains what functions in these 16 source code files allegedly implement the accused [redacted] Apple is not asking for expert reports ("how" a particular piece of source code shows infringement), but only for Maxell to give it fair notice of ***which*** software functions Maxell believes are relevant. D.I. 306 at 3.

II. Maxell's Grouping of Multiple Claim Elements Exemplifies Its Overbroad Citations

Maxell does not dispute that its SSIC cite source code across groups of claim elements. D.I. 284 at 4-5; D.I. 306 at 3-4. Maxell largely abandons its claim that these disparate claims are "similar," and instead presents a red herring—the source code files are "linked to each other." D.I. 313 at 3. All computer programs are built using source code files that call or link to each other. This does not justify Maxell's grouping of disparate claim elements to the same source code. P.R. 3-1(g) requires "an element-by-element" analysis, which cannot be satisfied by citing the same group of files across 22 claim elements of three patents.

III. Maxell Should Not Be Permitted To Continue To Hide the Relevant Source Code

Maxell's 70-page, multi-column source code appendices list over [redacted] in their entirety. D.I. 306 at 1. Maxell argues that this [redacted] number includes different versions of the same files listed across multiple operating systems. D.I. 313 at 4.[3] But different versions of a file across 7 operating system generations are not necessarily the same. By listing entire files, rather than identifying the specific accused functions in those files, Maxell has made it impossible to determine whether any differences between versions of the same file are material.

Maxell's reliance on the average number of files per patent/claim being only a dozen is misplaced. D.I. 313 at 4. But Maxell grouped multiple claim elements together, artificially

[3] Maxell complains that Apple has not agreed to representative products. D.I. 313 at 4. But Maxell chose to make this case as big as it is, and infringement remains Maxell's burden to prove. Moreover, at a minimum, Maxell's continuing failure to comply with P.R. 3-1(g) made it impossible for Apple to have agreed to representative products.

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**