

**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**



**APPLE INC.'S 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**

Maxell’s opposition (D.I. 299) confirms that it can comply with the Court’s prior orders requiring source code specificity but refuses to do so. Maxell’s strategy is clear—run out the clock and force Apple’s experts to disclose their invalidity opinions without the complete understanding of Maxell’s infringement theories P.R. 3-1(g) entitles them to have. Maxell presents a litany of excuses for why the Court should permit its noncompliance with P.R. 3-1(g) and this Court’s orders, trotting out familiar tropes about Apple’s discovery responses (disputes over which have been resolved largely in *Apple’s* favor), blaming “rigid source code restrictions” (to which Maxell agreed and which Apple relaxed at Maxell’s request), and alleging Apple moved too late (ignoring that Apple first raised this issue last November). Setting this hand-waving aside, Maxell essentially repeats the same argument the Court previously rejected—that the non-source code portions of its contentions are sufficiently clear to obviate the need for specific source code contentions. In so doing, Maxell misses the point of P.R. 3-1(g), which requires Maxell to identify “what source code of each Accused Instrumentality allegedly satisfies” claim elements it has identified as “software limitations.” And Maxell makes no effort to explain how its 70-page, multi-column source code appendices—blandly listing over [REDACTED] citations to complete source code files, many of them over 100 pages long—are “sufficiently focused to the accused functionality.” *See* D.I. 204 at 5.

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

**'493 Patent.** For the “mixing or culling” limitation of the '493 patent (element 5.e), Maxell’s SSIC cite [REDACTED] source code files in their entirety for just one accused product (iPhone 6S or “N71” running iOS 13).<sup>1</sup> *See* D.I. 284-1 at ¶ 6. But regardless of the number of files cited, the ultimate (and only) relevant question according to this Court’s prior order is whether Maxell

<sup>1</sup> Maxell misrepresents its own SSIC as citing only [REDACTED] files by ignoring the files listed on pages [REDACTED] of its “source code” table. D.I. 284-01 at ¶ 6. And because Maxell lists multiple versions of files across multiple operating systems, the total number of files listed across all operating systems is much greater.

has cited code with “some focus on the accused functionality” to provide “fair notice” of the same. D.I. 204 at 4. Here, Maxell has no response. It does not dispute that its SSIC cite well over 100 pages of source code for this single limitation, nor does it dispute that the cited files contain hundreds of different software functions. D.I. 284 at 3. And Maxell cannot point to anything in its SSIC that explains which of these myriad “processing functions relate to the accused functionality of ‘mixing or culling signal charges,’” as required. D.I. 204 at 3. Instead, Maxell rehashes the argument the Court previously rejected that the “non-source code evidence” in its P.R. 3-1(a) contentions is sufficiently clear. D.I. 299 at 3-4. But those do not sufficiently disclose what *source code* functionality Maxell accuses for the limitations it has *chosen* to designate as “software limitations”—that’s what P.R. 3-1(g) is for. Maxell argues, for example, that it limited element 5.e to “[REDACTED].” But its own citation shows that Maxell uses this phrase interchangeably with the claim term: “[REDACTED] [REDACTED].” D.I. 299 at Decl. ¶ 4. Maxell never explains how replacing “[REDACTED]” with “[REDACTED]” gives fair notice of which functions within the hundreds of cited source code pages are relate to the “culling” limitation.

As for claim element 5.d, Maxell does not dispute its SSIC list 18 source code files in their entirety for this single element. D.I. 284 at 3; D.I. 299 at 4. Maxell says it has limited this element to “one infringement theory,” but then cites a paragraph listing “[REDACTED].” D.I. 299 at Decl. ¶ 5. And Maxell does not explain how its SSIC correlate these 18 source code files (or any of their functions) to its existing infringement theories, as required by P.R. 3-1(g).

**’794 Patent.** Maxell does not dispute its SSIC cite 16 files for element 1.f and that just one of those files—“[REDACTED]”—spans nearly 200 pages and covers over 100 different functions and methods. D.I. 284 at 3. Maxell claims that it identified only “one theory” of

infringement for element 1.f relating to “[REDACTED]” of the accused products, but then says that this is merely “[REDACTED].” D.I. 299 at 5. Maxell’s SSIC offer no explanation which functions in these 16 source code files allegedly implement the accused “[REDACTED]” Maxell identifies in its brief or the other nebulous allegations identified in its SSIC.

Maxell acknowledges that it printed a single-page, [REDACTED] function from “[REDACTED]” that is “[REDACTED].” D.I. 299 at 5. Maxell’s ability to print a specific function from the 200-page file exemplifies the overbroad and noncompliant nature of Maxell’s P.R. 3-1(g) citations. A single source code function can take up less than a single page. D.I. 284 at 4. If Maxell intends to rely on a printed function, it must give fair notice of that rather than cite the entire 200-page file (along with 15 other files) without explanation. Although Maxell can identify allegedly relevant code for printing, it refuses to limit its SSIC to the printed pages or otherwise any provide information specifying the relevant ranges of source code. Apple is not asking for “expert report level detail,” which would presumably explain *how* the source code functions map to claim elements. Rather, Apple is asking Maxell to comply with P.R. 3-1(g) and simply identify *which* source code functions within that 200-page file (and other files) it believes “satisf[y] the software limitations of the asserted claim elements.” See D.I. 204 at 4.

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

The SSIC cite the same source code files across groups of claim elements, violating P.R. 3-1(g)’s requirement for “element-by-element” analysis. Maxell’s only excuse is that some of those elements have “nearly identical claim language” or “very similar functionality.” D.I. 299 at 5-6. The seven elements Maxell groups together for the ’794 patent—directed to everything from [REDACTED]

[REDACTED]

[REDACTED]—show that neither is the case.

D.I. 284 at 4. And unsurprisingly, Maxell’s grouping of 22 elements across the ’317, ’999, and ’498 patents include widely disparate elements. *Id.* at 5. Forcing Apple to look at product screenshots referred to in textual descriptions as some kind of “Rosetta Stone” to divine what functions in the 200-page “██████████,” 100-page “██████████,” and 20 other files are accused of infringement is not what P.R. 3-1(g) requires. *See* D.I. 284 at 5.

### **III. Maxell Should Not Be Permitted to Continue to Hide The Relevant Source Code**

Maxell has no intention to specifically disclose “what source code of each Accused Instrumentality allegedly satisfies the software limitations of the asserted claim elements.” P.R. 3-1(g).<sup>2</sup> Maxell treats this rule and the Court’s prior orders as mere suggestions. Indeed, just as Apple predicted (D.I. 123 at 5-6), Maxell now wants its experts to be able to rely on source code Apple produced before February 2020 not even specifically listed in its deficient SSIC. And Maxell admits that it can provide the specificity the Court’s order guarantees Apple if it chooses to—it disingenuously offered to provide this information for a single limitation out of the 20 asserted claims. *See* D.I. 299 at 2. Maxell cannot declare itself exempt from the even playing field the local patent rules are designed to create.

It is of no moment that Apple’s engineers can “understand the accused functionalities” and answer questions “regarding certain source code projects.” D.I. 299 at 6. It is unsurprising that Apple’s engineers—many of whom wrote the very source code at issue—can answer questions about printouts of selected source code functions. But P.R. 3-1(g) does not let Maxell burden Apple’s engineers with sorting through Maxell’s 3,700-page infringement contentions to guess what portions of source code are relevant to Maxell’s infringement theories. Being able to

---

<sup>2</sup> Maxell baldly argues that its failure to provide compliant contentions has “no bearing on invalidity.” D.I. 299 at 6. But it is beyond question that Maxell’s infringement theories directly impact Apple’s invalidity defenses. *See Upsher-Smith Labs. v. PamLab, LLC*, 412 F.3d 1319, 1322 (Fed. Cir. 2005) (“a product which would literally infringe if later in time anticipates if earlier”).

# 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.