

IN THE UNITED STATES DISTRICT COURT  
FOR THE DISTRICT OF DELAWARE

PERSONAL AUDIO, LLC,

Plaintiff,

v.

Civil Action No. 17-1751-CFC

GOOGLE LLC,

Defendant.

---

Brian E. Farnan, Michael J. Farnan, Rosemary J. Piergiovanni, FARNAN LLP,  
Wilmington, Delaware; Steven M. Hanle, Jason de Bretteville, Douglas Q. Hahn,  
Salil Bali, Lisa Northrup, Ahmad Takouche, Henning Schmidt, STRADLING  
YOCCA CARLSON & RAUTH, P.C., Newport Beach, California


*Counsel for Plaintiff*

Jack B. Blumenfeld, Brian P. Egan, Cameron P. Clark, MORRIS, NICHOLS,  
ARSHT & TUNNELL LLP, Wilmington, Delaware; Melissa J. Baily, David A.  
Perlson, Antonio R. Sistos, Jeff Nardinelli, QUINN EMANUEL URQUHART &  
SULLIVAN, LLP, San Francisco, California; Patrick Stafford, QUINN  
EMANUEL URQUHART & SULLIVAN, LLP, Washington, District of  
Columbia; Owen F. Roberts, QUINN EMANUEL URQUHART & SULLIVAN,  
LLP, New York New York; Olga Slobodyanyuk, QUINN EMANUEL  
URQUHART & SULLIVAN, LLP, Redwood Shores, California; Lance Yang,  
QUINN EMANUEL URQUHART & SULLIVAN, LLP, Los Angeles, California

*Counsel for Defendant*

**MEMORANDUM OPINION**

September 5, 2023  
Wilmington, Delaware



---

COLM F. CONNOLLY  
CHIEF JUDGE

Plaintiff Personal Audio, LLC (Personal Audio) sued Defendant Google LLC (Google) for infringement of claims 3, 6, and 13 of U.S. Patent No. 6,199,076 (the #076 patent) and claims 7 and 12 of U.S. Patent No. 7,509,178 (the #178 patent). The asserted patents cover an audio program player that automatically plays a predetermined schedule of audio program segments—songs, for example—from a program library. The claimed player also allows listeners to use commands (e.g., “skip”) to alter the sequence and content of the audio program segments. Personal Audio alleges that Google’s “Google Play Music” (GPM) software, which was installed on Google devices (e.g., Google’s Pixel C tablet) and third-party devices (e.g., unlicensed third-party cell phones with GPM installed), infringes the asserted patents.

After a six-day trial, the jury found Google liable for direct and induced infringement of claims 3 and 6 of the #076 patent and claims 7 and 12 of the #178 patent. It also found that Google willfully infringed or willfully induced users of unlicensed Android phones with GPM installed to infringe those claims and that all the asserted claims are not invalid. The jury awarded Personal Audio \$15.1 million in damages.

Pending before me is Google’s Renewed Motion For Judgment As A Matter Of Law Or New Trial (D.I. 863).

**I. MOTION FOR JUDGMENT AS A MATTER OF LAW (JMOL)**

**A. Legal Standard**

“If the court does not grant a motion for judgment as a matter of law made under Rule 50(a), . . . the movant may file a renewed motion for judgment as a matter of law and may include an alternative or joint request for a new trial under Rule 59.” Fed. R. Civ. P. 50(b). “The grant or denial of a JMOL motion is a procedural issue not unique to patent law, reviewed under the law of the regional circuit in which the appeal from the district court would usually lie.” *TI Grp. Auto. Sys. (N. Am.), Inc. v. VDO N. Am., L.L.C.*, 375 F.3d 1126, 1133 (Fed. Cir. 2004).

A party that does not have the burden of proof is entitled to a judgment as a matter of law “only if, viewing the evidence in the light most favorable to the nonmovant and giving it the advantage of every fair and reasonable inference, there is insufficient evidence from which a jury reasonably could find liability.” *Lightning Lube, Inc. v. Witco Corp.*, 4 F.3d 1153, 1166 (3d Cir. 1993).

**B. Analysis**

**1. Direct Infringement**

Google argues that Personal Audio “did not adduce substantial evidence that the accused products satisfied the [asserted claims’] ‘sequencing file’ limitations or

the [asserted claims'] means-plus-function elements [that were] construed to require 'LocType' algorithms." D.I. 863 at 2.

**a. "Sequencing File" Terms**

**1) Relevant Pretrial Proceedings**

The claim term "sequencing file" appears in each of the #178 patent's asserted claims, and the term "file of data establishing a sequence" appears in each of #076 patent's asserted claims. The parties referred to these terms as the "sequencing file" terms, and I construed "sequencing file" to mean "a file that is received by the player, stored, and used by the processor to both control playback of each song in the ordered sequence and respond to control commands." D.I. 447 at 2.

Before trial, Google filed a "Motion in Limine #2 – Preclude Argument That More Than One File Can Meet The 'Sequencing File' Limitation." D.I. 799-33 at 2. Google stated in the motion that Personal Audio "intend[ed] to offer [trial] testimony and argument that the asserted claims can be met when one sequencing file is downloaded and stored but another file is used to control playback." D.I. 799-33 at 4.

When the parties presented oral argument on the motion in limine during the pretrial conference, I asked Personal Audio's counsel: "At the end of the day, are you going to acknowledge in front of the jury or rather are you going to assert [in

front of the jury] that there was a sequencing file, a single one that performed all three functions?” D.I. 823 at 128:10–13. Personal Audio’s counsel responded: “We are going to do that for sure.” D.I. 823 at 128:14. I took Personal Audio’s representation at face value and said to Google’s counsel:

So, Google, here’s the thing. If they are going to hold themselves to [my claim construction of “sequencing file”] -- they’ve got to prove that there’s at least one sequencing file that executed all three functions at some point. [And] they said -- they just said that’s what they are going to prove.

D.I. 823 at 129:1–5.

Google, however, did not take Personal Audio’s representation at face value.

In response to my comments, Google’s counsel said:

Your Honor, but the way that they’re going to do it -- this is the problem. This is why we have to keep coming back to this -- they say that, but the way they intend to do it is to go against exactly what they argued at claim construction. *They are going to say that the file that was received and stored is used because you extract the information from it and put it into another file that controls playback.*

That is their theory under equivalence. They said it in their own opposition. They say in their own opposition, Page 3 -- the issue of whether the claims as so defined exclude the use of multiple sequencing files was not at issue.

*They want to say that they extract -- the way Personal Audio gets around what was clearly decided and intended to be decided through claim construction is they say, look, the downloaded and stored file is used for*

*everything because the information is extracted from it, put into different data structures, and those data structures control playback.*

So originally, the received file was used because the sequence was taken from the received file. That's how they are saying that they are going to argue their case. But if you look . . .

-- that's exactly what they were arguing at claim construction. They were arguing against our claim construction because they wanted to argue exactly that. [Personal Audio] wanted to argue that the sequence from that received sequencing file to be used to respond to control playback can be copied into another file that is used to control playback. *So their use of the original file is: Well, we used it. We used it to take the information and copy it into other files that then control playback.* But that was rejected at claim construction. That was the whole argument at claim construction.

D.I. 823 at 129:6–30:16 (emphasis added).

As it turned out, Google's prediction about how Personal Audio would attempt to prove at trial the existence of a "sequencing file" in GPM was spot-on. But based on what I understood Personal Audio to say at the pretrial conference, I denied the motion in limine:

. . . . I didn't say, when I articulated the basis for my claim construction hearing, and having gone back and looked at the lexicography in question, *I don't believe it's the case that the use of the file precludes the use of another file to execute the control function.*

*All I said was that the file had to be used to control it, but that doesn't mean it can't be used with something else. But it must be used as part of the control.*

So I'm going to deny the motion in limine.

D.I. 823 at 130:20–31:4 (emphasis added).

Five days before trial began, Google filed a motion for “clarification” of the “sequencing file” construction. D.I. 834. Google stated that it understood my comments from the pretrial conference to mean that the “sequencing file” construction “requires the processor to use the received sequencing file when controlling playback of each song and when responding to control commands, even though the processor may also use ‘something else’ in addition to the received sequencing file when controlling playback of each song.” D.I. 834 at 3. But Google remained concerned that Personal Audio was going to argue that “infringement occurs (either literally or by equivalents) when one file is received and stored, *some information from that file is copied into a second file*, and *only* that second file is used to control playback of each song and respond to control commands.” D.I. 834 at 3 (some emphasis added). Google therefore “requeste[d] clarification that the Court’s claim construction of ‘sequencing file’ precludes infringement where (i) one ‘sequencing file’ is received and stored, (ii) *some information from that ‘sequencing file’ is copied into a second file*, and (iii) the processor uses *only that second file to control playback* of a song or respond to control commands.” D.I. 834 at 4 (some emphasis added). It also proposed a clarifying construction to be presented to the jury. D.I. 834 at 5.

Personal Audio filed an answering brief in which it argued that “[a]s long as the file is used to control playback of each song and to respond to commands, it meets [the] limitation regardless of the timing of that use by the processor to control playback.” D.I. 839 at 3. Personal Audio insisted that my construction does not prevent “the access to and/or extraction of data from the downloaded sequencing file to control playback [a]s a use of the downloaded file for” controlling playback. D.I. 839 at 6. But it also stated: “It is beyond reasonable dispute that accessing and extracting information from the downloaded sequencing file to load other *control* structures is using the downloaded sequencing file as part of the control.” D.I. 839 at 6 (emphasis in the original; internal quotation marks and citation omitted).

After reading the parties’ briefs, I did not change my construction of “sequencing file.” But I was concerned by Personal Audio’s statement that “accessing and extracting information from the downloaded sequencing file to load other control structures is using the downloaded sequencing file as part of the control” because that statement suggested that Personal Audio would attempt to argue during trial that copying the sequencing file and using only that copy to control playback would not violate my claim construction. I therefore issued an order that stated in pertinent part:

. . . [M]y claim construction of the “sequencing file” limitation requires that the file is (1) received by the



player, (2) stored, *and* (3) used by the processor to control playback of each song in the ordered sequence and respond to control commands. A theory of infringement based on a showing that only a second file is used to control playback of each song in the ordered sequence and respond to control commands is not consistent with my claim construction.

A copy of a file is not the file by rather is a new and distinct file (i.e., a second file). Thus, the use of only data from a copy of a first file constitutes the use of only a second file and does not constitute the use of the first file.

D.I. 843 at 1 (emphasis in the original). It was therefore crystal clear by the time trial began that in order to prove direct infringement of the “sequencing file” limitation, Personal Audio had to establish that an individual file was (1) received by the player, (2), stored, *and* (3) used by the processor to control playback of each song in the ordered sequence and respond to control commands. Evidence that only a copy of the received and stored file (or a part of that file) was used to control playback would not satisfy the “sequencing file” limitation.

## **2) Relevant Trial Proceedings**

At trial, Personal Audio primarily elicited evidence about the “sequencing file” terms through its infringement and invalidity expert, Dr. Kevin Almeroth. Dr. Almeroth testified that “the” sequencing file is initially *stored* in one data structure: the LISTITEMS Table. In his words:

[T]here’s a server that’s providing what’s called JSON data, Java Script Object Notation. That was part of the

code that I showed for downloading the sequencing file. And then it was stored -- the sequencing file was stored into the list items table. I showed the source code for the list items table.

. . . . [T]hat list items table contains all of the lists, all of the songs that are in lists, and then their positions within that respective list.

Tr. of June 12–20, 2023 jury trial at 490:24–91:8. Dr. Almeroth also testified that the sequencing file is *accessed* through the “list ID”:

And the way that the sequencing file is accessed in that list items table is through the list ID. So you see the list ID come across from the JSON object at the server. And it’s in the list items table. And so that’s the name by which the list -- the sequencing file would be accessed.

Tr. 491:9–14. He further testified that the data stored in the LISTITEMS Table are *moved* into another data structure, the QUEUE\_ITEMS TABLE, at which point they become the “sequencing file”:

. . . . [T]hose three items that are part of the playlist, *are then moved* into the queue items table. And so now you have the same songs in -- . . . those would be the three songs that were part of the list 2481.

Now, the name for this data structure is the queue items table. And so that’s how the sequencing file is referenced when you have the queue items table.

The position now becomes specific to the particular music ID and its location in the queue items table. *So that’s, now, the sequencing file.*

Tr. 492:3–14 (emphasis added). And finally, Dr. Almeroth testified about how the data in the QUEUE\_ITEMS TABLE are “loaded into” a third data structure, the Queue, which becomes the “sequencing file”:

Okay. Then the final step is the queue items table. That information is loaded into the queue. And so *now the queue becomes the sequencing file*. The queue is now the name of that collection of data. *And it's the queue that's used for the user control*.

Tr. 493:10–14 (emphasis added).

In describing these steps, Dr. Almeroth testified that he was “[t]racing the collection of data through the system as -- and then that is the sequencing file as it moves through the system.” Tr. 493:6–8. He summarized his opinion as follows:

So the sequencing file starts in the JSON, as a JSON object, comes across the network, is stored in the database in the list items table and then into the queue items table, then is stored in the queue. And so I believe Google's position is that that's not the same sequencing file because it changes location or different formats. I won't try and say exactly what they're going to say, but *what I will say is applying the construction for the term “file,” and for sequencing file, the fact that it moves across the network into the database and then is loaded into the queue, does not mean that there is not infringement*.

Tr. 493:24–94:10 (emphasis added).

In its cross-examination of Dr. Almeroth, Google elected not to confront him directly about his testimony that the data he identified as the “sequencing file” *move* from the LISTITEMS Table to the QUEUE\_ITEMS TABLE and eventually

to the Queue. Google also chose not to confront Dr. Almeroth on cross about the Court's construction of "sequencing file" or the fact that the use of only a copy of the file received and stored by the processor (or a part of that file) to control playback would not satisfy that claim construction. In the middle of the cross, however, Google elicited the following testimony:

Q. You talked a bit about the list items table earlier?

A. Yes, ma'am, I did.

Q. Every entry in the table in the list items table has three fields of data, correct?

A. Yes, ma'am.

Q. And those three fields are "Playlist ID, Music ID, and Position," correct?

A. Yes, ma'am.

Q: *Now, when the queue items table is loaded, nothing about the list items table changes, does it?*

A. *That's correct.*

Q. *And when the queue is loaded, nothing about the queue items table changes, correct?*

A. *That's correct.*

Q. And list items is in a different memory location from queue items, correct?

A. It is.

Q. And queue items is in a different memory location from queue, correct?

A. It is.

Tr. 535:21–36:15 (emphasis added).<sup>1</sup>

The next day, Google timely moved for judgment as a matter of law of noninfringement under Rule 50(a). *See* D.I. 848; Tr. 1070:24–72:24. I denied the motion without prejudice to renew, saying “We’re going to see what the jury says, and then we’ll deal with all this after trial.” Tr. 1072:20–74:19.

The next day, during closing arguments, Personal Audio’s counsel made the following statement to the jury:

Google’s [s]econd argument [for why the jury should return a verdict of no direct infringement] is that we don’t have a sequencing file because we have a bunch of different files. And they showed you some slides in their opening that said -- that tried to distinguish between this data structure over here on the left, which is a JSON object that is downloaded; a list items table that is over here on the right, which is what it goes to when it’s on the player. There is the list items table.

And then it goes to the queue items table with the same collection of data flowing through and then from

---

<sup>1</sup> The logically unavoidable conclusion from this testimony is that the data that Dr. Almeroth says are “loaded” into the QUEUE\_ITEMS\_Table and eventually into the Queue are copied from data received and stored in the LISTITEMS Table. It is understandable, however, that the jury would have overlooked this subtle (though substantively effective) line of questioning, especially since it comprised less than one page of the 35 pages of Dr. Almeroth’s cross-examination and was not highlighted by an introductory transition statement or other means during the cross or mentioned by Google in closing argument.

the queue items table to the queue. So it's all the same collection of data that -- and this is according to the Court's claim construction -- that same collection of data that flows through the Google Play Music application.

*And it leaves behind the same data in the previous data structure, but it's the same collection of data that moves through, and it's received by the player and stored by the processor to both control playback of songs and to respond to control commands like skip forward and skip back.*

Tr. 1124:13–25:7 (emphasis added).

### **3) Google is Entitled to JMOL**

Dr. Almeroth's testimony, viewed in the light most favorable to Personal Audio, does not provide a sufficient evidentiary basis for a rational juror to find that Google's GPM has a "sequencing file." Again, I construed "sequencing file" to mean "*a file* that is received by the player, stored, and used by the processor to both control playback of each song in the ordered sequence and respond to control commands." D.I. 447 at 2 (emphasis added). That means that a single file must be received, stored, and used to both control playback and respond to commands. Dr. Almeroth, however, testified that one file—the LISTITEMS Table—is received by the player and stored, and that a *different* file—the Queue—is "used for the user control." See Tr. 490:24–91:8; 493:10–14.

Although Dr. Almeroth testified during his direct examination that the data in the LISTITEMS Table that he identifies as the sequencing file are *moved* into

the QUEUE\_ITEMS Table and then the Queue, he admitted on cross that “nothing about the list items table changes” when the data are loaded into the QUEUE\_ITEMS Table and the Queue. Tr. 536:4–6. To use Personal Audio’s counsel’s words, “the same collection of data that moves” into the QUEUE\_ITEMS Table and the Queue is “le[ft] behind” in the LISTITEMS Table. Tr. 1125:2–7. In other words, Dr. Almeroth’s testimony necessarily means that the data in the LISTITEMS Table are *copied*, and *the copied data* are then moved into the QUEUE\_ITEMS Table and the Queue. As made clear before trial, however, that infringement theory does not satisfy the “sequencing file” claim construction. Accordingly, Google is entitled to JMOL of noninfringement. *See Laitram Corp. v. Rexnord, Inc.*, 939 F.2d 1533, 1535 (Fed. Cir. 1991) (“[T]he failure to meet a single limitation is sufficient to negate infringement of the claim.”).

Personal Audio argues in its posttrial briefing that “undisputed evidence presented at trial [showed] that LISTITEMS is also used [with the Queue] to control playback, and is thus a ‘sequencing file’ under the Court’s construction.” D.I. 868 at 1–2 (emphasis in the original). In support of this argument, Personal Audio points to the testimony of “GPM engineer Razumeiko and Google’s infringement expert [Dr. Mayer-Patel] . . . that each time playback is initiated (with a ‘play’ or ‘go’ command), the GPM processor directly references and therefore uses the playlist in LISTITEMS to seed the queue and to commence playback.”

D.I. 868 at 2 (citations omitted; emphasis in the original). But at most, this testimony shows that the LISTITEMS Table takes part in initiating playback of the first song in a playlist; and, as discussed above, the received and stored “sequencing file” must “control playback of *each song* in the ordered sequence *and* respond to control commands.” D.I. 447 at 2 (emphasis added).

Personal Audio next argues that “the playlist in LISTITEMS is used to control the order that the songs used for playback.” D.I. 868 at 2 (emphasis in the original; internal quotation marks omitted). But the testimony it cites in support of this assertion establishes only that a copy of the LISTITEMS playlist moved into the Queue controls playback. *See* Tr. 336:15–17 (John Evans, a Google engineering director, testifying by deposition that “[t]he order of the playlist would be the order of the queue when it is created, and thus, the order that the queue is played back in”); Tr. 586:6–9 (Mr. Razumeiko agreeing with Mr. Evans). The source code comments that Personal Audio says “further confirm that the playlist is used to control playback,” D.I. 868 at 2 n.1 (emphasis in the original), also prove otherwise. The comments state that GPM “*replaces* the current playlist with a *new list* and starts playing it at the specified position in the list.” Tr. 588:20–22 (Mr. Razumeiko reading GPM source code) (emphasis added). *Replacing* the current playlist means that *a different* sequencing file—i.e., a copy—is being used during playback.



Personal Audio also argues in its briefing that the LISTITEMS Table “controls playback of each song in the ordered sequence” because GPM “continually references” a code function called “listitemsid” that Personal Audio says is “contained in [the] LISTITEMS [Table]” “to confirm that each song in the ordered sequence is playable.” D.I. 868 at 3 (emphasis in original). But the record evidence cited by Personal Audio, *see* D.I. 868 at 3, does not support this assertion. Indeed, no witness testified at trial about what a “listitemsid” is or where it is “contained.” And in any event, even if a “listitemsid” code function were contained in the LISTITEMS Table (or within a file within that Table) and controlled the playback of each song in an ordered sequence, Personal Audio does not contend, and there is no evidence from which to conclude, that this code function (or the LISTITEMS Table) otherwise “respond[s] to control commands” as required by the construction of “sequencing file.”

Finally, Personal Audio argues in its posttrial brief that the evidence it discusses in its brief “supports Cases 1–3 that the Court already found, if proven, would meet the sequencing file construction” and that “Google’s [JMOL] motion ignores Cases 1, 2, and 3 and attacks [Dr.] Almeroth’s explanation of how the data from the LISTITEMS Table is used to populate the queue in support of Use Case 4.” D.I. 868 at 4–5 (emphasis in the original). To be blunt, although I think Personal Audio is making here a claim construction argument, I really have no idea

what it is talking about. Personal Audio does not define the “Cases” it has in mind, let alone discuss how they are relevant to the issue at hand. It cites in support of this argument a Memorandum Order (D.I. 720 at 3) issued by the Magistrate Judge earlier in this action that refers to “cases 1, 2, and 3” in a parenthetical to a citation to a reply brief (D.I. 630 at 3) Google filed in support of a *Daubert* motion to exclude Personal Audio’s infringement expert’s testimony about “sequencing file.” “Judges are not like pigs, hunting for truffles buried in briefs.” *United States v. Dunkel*, 927 F.2d 955, 956 (7th Cir. 1991). Accordingly, Personal Audio has forfeited whatever arguments it purported to make about “Cases” 1 through 4.

In sum, viewing all the record evidence in the light most favorable to Personal Audio and giving it the advantage of every fair and reasonable inference, there is insufficient evidence from which a jury could reasonably find direct infringement of the asserted claims’ “sequencing file” limitation. Accordingly, I will grant Google’s JMOL motion with respect to direct infringement.

**b. “LocType” Terms**

Each of the asserted claims has a means-plus-function limitation that requires a sequencing file with “LocTypes.” *See* D.I. 863-1. Google argues that Personal Audio did not adduce sufficient evidence for the jury to find that the accused products had the structural equivalent of a LocType.

A claim limitation that recites a function to be performed rather than a definite structure is subject to the requirements of 35 U.S.C. § 112, ¶ 6 (1994). *Odetics, Inc. v. Storage Tech. Corp.*, 185 F.3d 1259, 1266 (Fed. Cir. 1999). Such a limitation “must be construed ‘to cover the corresponding structure, material, or acts described in the specification and equivalents thereof.’” *Id.* at 1266–67 (citing 35 U.S.C. § 112, ¶ 6; *see also B. Braun Med., Inc. v. Abbott Labs.*, 124 F.3d 1419, 1424 (Fed. Cir. 1997)). “Literal infringement of a § 112, ¶ 6 limitation requires that the relevant structure in the accused device perform the identical function recited in the claim and be identical or equivalent to the corresponding structure in the specification.” *Id.* If the relevant structure in the accused device is not identical to the corresponding structure in the patent’s written description, then the test for § 112, ¶ 6 equivalence is whether the two structures “perform the identical function, in substantially the same way, with substantially the same result.” *Kemco Sales, Inc. v. Control Papers Co.*, 208 F.3d 1352, 1364 (Fed. Cir. 2000). Because “[f]unctional identity *and* either structural identity or equivalence *are both necessary*,” *Odetics*, 185 F.3d at 1267 (some emphasis added), a court is required “to give independent meaning to both the ‘function’ and ‘way’ prongs of the equivalency test.” *Applied Med. Res. Corp. v. U.S. Surgical Corp.*, 312 F. App’x 326, 332 n.3 (Fed. Cir. 2009).

**1) “Means for Continuously  
Reproducing/Delivering” Terms**

I construed the corresponding structure for the asserted claims’ “means for continuously reproducing/delivering” terms to include an algorithm that performs the following steps:

1. beginning playback with the program segment identified by the ProgramID contained in the Selection\_Record specified by the CurrentPlay variable;
2. when the currently playing program concludes, incrementing the CurrentPlay variable by one and fetching and playing the program segment identified by the ProgramID contained in the next Selection\_Record in the sequencing file;
3. repeating step (2) until a rewind Selection\_Record (LocType: R) in the sequencing file is reached, which resets the CurrentPlay variable to the location value contained in the rewind Selection Record which is set to “1” to begin the playing sequence again with the first Selection\_Record in the sequencing file.

D.I. 863-1 at 3–4. Dr. Almeroth described these three steps as follows:

So the first one is the beginning playback with the program segment identified by the program ID contained in the selection record specified by the CurrentPlay variable.

So you’re beginning playback of one song. That’s kind of step one. The second step is when the currently playing program concludes, incrementing the CurrentPlay variable by one and fetching and playing the program segment identified by the program ID contained in the next selection record in the sequencing file.

So, in other words, increment the counter to look at the next record to see what the next song in the playlist is. Okay?

And then the third step is you keep going to the next song in the playlist until you hit a rewind selection record LocType of "R" in the sequencing file. And what that does is it resets the CurrentPlay variable to the location value contained in the rewind selection record, which is one. So it goes back to the beginning of the playlist. And that's what the steps of the structure for this limitation require.

Tr. 497:1–21.

Dr. Almeroth also testified that the differences between these steps and GPM's source code, like using a mechanism besides a LocType "R," were not substantial structural differences:

. . . . So these kinds of structures allow for structural equivalents. So for example, if the name of the variable is not "program ID" or "CurrentPlay," that's not a particularly big difference.

If the way that you implement the check that your end of the list is not with a LocType "R" but some other mechanism, that wouldn't be a significant difference.

Tr. 497:24–98:5; *see also* Tr. 499:12–19 ("So let me summarize. So the requirement here is that the differences between the accused product and the patent are insubstantial. And so the way that you can test for -- that it's not a substantial difference is if it's interchangeable, you can use one versus the other in that

scenario, or that you're performing the same function in essentially the same way, substantially similar, to get to the same result.") (Almeroth).

Personal Audio argues that in addition to the above testimony, Dr. "Almeroth provided detailed step-by-step testimony that the 'way' GPM implements continuous playback—as a whole—performs the recited function in substantially the same way as the structure in the Court's construction, and that they achieve substantially the same result of continuous playback in a loop." D.I. 868 at 10. There is record evidence that supports this contention. Dr. Almeroth testified, for example, about GPM's "set next track function": the steps that GPM processes "to figure out what the next song is that should be played." Tr. 501:3–18. That function includes "a check at 5797 that checks whether or not you're at the end of the playlist. And so the way that it checks is to say, is the current player position greater than or equal to the length of the playlist." Tr. 502:4–7. Dr. Almeroth expounded:

Okay. So in nonmathematical terms, it says, am I on the last song or not? And if I'm on the last song, what it sets it to is back to the first song. And that happens -- you see case at 5799. That's "repeat current." That would get executed if you're just repeating one song. And then at 5811, that's the case for "repeat none." So no repeat, you would stop.

And the case that we care about is at 5817 where you "repeat all," and then you return the target position. And the target position was calculated -- sorry to jump around -- but at 5795, target position equals the play

position essentially divided by the length. And so that sets the target position to one. It checks when you're at the end. *If you're at the end, it sets it back to one. . . .*

The context here is we're talking about if you continue to play songs and you have to find the next one, it keeps playing the next song, always checking to see if you're at the end of the list. *And if you are, you circle back to the first song.*

Q. And after reviewing all of the code in the operation of the app, did you conclude, based on your analysis, that the Google Play Music performs at algorithm [sic] as construed by the Court in the same -- as required by the claim element?

A. I determined that it meets it under the same or equivalent structure as defined in the construction.

Tr. 502:8–03:14 (emphasis added).

Dr. Almeroth also testified about Google's litigation position that GPM's structure is not structurally equivalent because it "does not use LocTypes":

Q. And so do you agree with -- with Google's argument that they don't meet this limitation because they don't use LocTypes?

A. No. I disagree.

Q. And why is that?

A. Well, for two reasons. One, the idea it has to match with [a figure in the patents' written descriptions that described an embodiment] is not correct. That's not the test. It's whether it meets the limitation under the Court's construction. And I showed what that algorithm was. It includes a LocType, but I don't believe Google's analysis accounts for structural equivalence. They don't have to

be the same variable names, and the technique can be substantially similar and still meet the limitation.

So even though Google Play Music checks that you're at the end of the list based on the number of list items and circles back based on that, I believe that structure is insubstantially different from the requirements of the claim construction.

Tr. 504:11–05:3.

Viewing the above evidence in the light most favorable to Personal Audio and giving it the advantage of every fair and reasonable inference, there is sufficient evidence from which a jury could reasonably find direct infringement of the LocType limitation in the “means for continuously reproducing/delivering” terms.

## **2) “Means for Responding to a Skip Command” Terms**

Claims 3 and 6 of the # 076 patent and claim 7 of the #178 patent include a “means for responding to a skip command” limitation, which I construed to require an algorithm that performs the following steps:

1. Scanning forward in the received sequencing file to locate the next Selection\_Record of the appropriate LocType;
2. Resetting the CurrentPlay variable to the record number of that Selection\_Record; and
3. Fetching and playing the program segment identified by the ProgramID contained in the new Selection\_Record.

A LocType is a single byte character and an identifier that indicates a characteristic of a selection record.



D.I. 873-1 at 2, 5. Dr. Almeroth described those steps in the following manner:

It says, “Scanning forward in the received sequencing file to locate the next Selection\_Record of the appropriate LocType.” Then you reset “the CurrentPlay variable to the record number of the Selection\_Record,” the next one that was identified. And then you fetch and play the program segment identified by the program ID.

So as you can probably anticipate, the dispute is whether or not there is an appropriate LocType in Step 1, or the structural equivalence of the entire algorithm.

Tr. 506:18–07:2; *see also* Tr. 507:18–20 (“As I said, the real dispute is whether or not there is a LocType that’s required under structural equivalence.”) (Almeroth).

Google argues that Personal Audio has not adduced sufficient evidence to support the jury’s verdict because the “Music IDs” used in GPM are not structurally equivalent to LocTypes. D.I. 863 at 17. Personal Audio responds that it adduced, through Dr. Almeroth, “step-by-step evidence as to the ‘way’ GPM’s code operates as a whole to perform the recited function in substantially the same way as the claimed structure and achieves the same or substantially the same result of locating the next song and playing it.” D.I. 868 at 7 (emphasis in the original).

It points, for instance, to the following testimony from Dr. Almeroth:

. . . . [T]he code that that calls will actually look at the next song in the sequence and determine if it’s playable. And if it’s not, it will find the next song after that that’s playable.

So, for example, why would a song in a playlist not be playable? And the answer is a couple of reasons. So one is that you're in airplane mode and this particular song hasn't been downloaded. So it uses the music ID and then tries to find where that song is stored. And so if the music ID is not playable, then it will move to the next item in the list.

So it actually scans forward in that sequencing file to find the next song that's playable, and it uses the music ID to determine if that next song is playable or not.

\* \* \* \*

So this ties into that first step. Scanning forward in the sequencing file to locate the next selection record of the appropriate LocType.

In the code, what it does is it uses the music ID and its equivalent to the LocType to determine whether or not that song is appropriate, whether or not it's playable. So in that instance, the music ID is the LocType.

Tr. 510:1–11:3. Dr. Almeroth also testified about why a music ID is structurally equivalent to a LocType despite not being a single byte character:

Now, at the bottom, it says, "The LocType is a single byte character," and in the examples, it's not a single byte. A byte is eight bits. That only gets you numbers between about zero and 256. So in Google Play Music, it's not one byte; it's multiple bites. Well, that's a structural equivalence.

The key to the invention is not having a single byte or two bytes. The novelty of the invention is really the program player with the advanced functionality. That's why there are structural equivalents, so that these minor differences in how this algorithm is implemented doesn't take a system outside of the scope of the claims.

So that's why I think in this instance, using the music ID is a structural equivalence to what this claim requires.

Tr. 511:6–20; *see also* Tr. 512:2–18. Dr. Almeroth further responded to Google's argument that a music ID is not structurally equivalent to a LocType:

Q. . . . Okay. So what -- does Google agree with your opinion that Claim 4 is performed by Google Play Music?

A. No, they don't.

Q. And what is your understanding of what they dispute?

A. It's a very similar reason as previously. They say that there has to be absolutely be a LocType just like exactly the algorithm requires. I don't believe they're allowing for structural equivalence, so that's why I disagree with that conclusion.

Q. Okay. And you talked earlier about whether particular LocTypes that are in the written description of the patent, like an "R" or an "S" or a "T," are required in the Court's claim construction. What was your opinion?

A. That they were not. And for this limitation, there's no identification of a specific LocType, and so music ID would fit just fine.

Q. So what would be the -- according to the source code you reviewed, what would be the appropriate LocType or its equivalent in the code?

A. The next playable music ID.

Q. So after hearing Google's argument and considering it and analyzing the source code and analyzing the app, is it your opinion still that Google Play Music on an accused device meets all the limitations and performs the algorithm of Claim 4 of the [#]178 patent?

A. Yes, that's correct.

Tr. 513:7–14:7; *see also* Tr. 517:2–9 (“I believe Google’s non-infringement positions are consistent with respect to either not having a LocType or having a LocType that does something different. And so the same analysis and conclusions and the discussion of my responses to their non-infringement opinions would apply to all of the other limitations that have been construed with the LocType term or the sequencing file in the term or the construction.”) (Almeroth).

Viewing this evidence in the light most favorable to Personal Audio and giving it the advantage of every fair and reasonable inference, there is sufficient evidence from which a jury could reasonably find infringement of the LocType limitation in the “means for responding to a skip command” terms.

\* \* \* \*

To sum up: I find that Personal Audio has not adduced sufficient evidence to support the jury’s direct infringement verdict with respect to the “sequencing file” limitations. Personal Audio has, however, adduced sufficient evidence to support the jury’s direct infringement verdict with respect to the “LocType” limitations. The “sequencing file” limitation is present in each of the asserted claims. Personal

Audio therefore has not adduced sufficient evidence to support the jury's direct infringement verdict, and accordingly, I find that Google is entitled to JMOL of no direct infringement.

## **2. Induced Infringement, Willfulness, and Damages**

Google also asks for JMOL of no induced infringement, no willfulness, and a lower damages award. D.I. 863 at 18, 21, 22. Because I will grant Google's motion for JMOL of no direct infringement, I need not address these arguments. "[T]here can be no inducement or contributory infringement without an underlying act of direct infringement." *In re Bill of Lading Transmission & Processing Sys. Pat. Litig.*, 681 F.3d 1323, 1333 (Fed. Cir. 2012) (citation omitted). Direct infringement is also "necessary, but not sufficient, for a finding of willfulness." *Bayer Healthcare LLC v. Baxalta Inc.*, 989 F.3d 964, 988 (Fed. Cir. 2021). And with no direct infringement, Personal Audio is not entitled to compensatory or enhanced damages. *See* 35 U.S.C. § 284 ("Upon finding for the claimant the court shall award the claimant damages adequate to compensate for the infringement . . . . [T]he court may increase the damages by up to three times the amount found or assessed.").

## **II. CONDITIONAL MOTION FOR A NEW TRIAL**

Under Rule 50(c)(1), "[i]f the court grants a renewed motion for judgment as a matter of law, it must also conditionally rule on any motion for a new trial by

determining whether a new trial should be granted if the judgment is later vacated or reversed.” Fed. R. Civ. P. 50(c)(1). In this case, were the Federal Circuit to vacate the judgment of no direct infringement, I believe a new trial would be warranted because, as explained above, Personal Audio effectively ignored the Court’s claim construction of “sequencing file” and because the jury’s verdicts with respect to direct infringement of the asserted claims of the #076 and #178 patents are contrary to the evidence.

### **III. CONCLUSION**

For the reasons discussed above, I will enter a judgment of no infringement of the asserted patents as a matter of law. I will also conditionally grant Google’s motion for a new trial under Federal Rule of Civil Procedure 50(c)(1).

The Court will issue an Order consistent with this Opinion.