Pages 1 - 77

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

BEFORE THE HONORABLE WILLIAM H. ALSUP

ORACLE AMERICA, INC.,	)	
	)	
Plaintiff,	)	
	)	
vs.	)	NO. C 10-03561 WHA
	)	
GOOGLE INC.,	)	
	)	San Francisco, California
Defendant.	)	Wednesday
	)	April 20, 2011
	)	1:28 p.m.

TRANSCRIPT OF PROCEEDINGS

APPEARANCES:

For Plaintiff:
Morrison & Foerster
755 Page Mill Road
Palo Alto, CA 94304-1018
(650) 813-5600
(650) 494-0792 (fax)
BY: Michael A. Jacobs
Marc David Peters
Roman Swoopes

For Plaintiff:
Oracle
500 Oracle Parkway M/S 50P7
Redwood Shores, CA 94065
(650) 506-9432
(650) 506-7114 (fax)
BY: Andrew C. Temkin

(Appearances continued on next page)

Reported By: Lydia Zinn, CSR #9223, RPR
Official Reporter - U.S. District Court

APPEARANCES (CONT'D)

For Defendant: King & Spalding LLP
1185 Avenue of the Americas
New York, NY 10036-4003
(212) 556-2100
BY: **Scott T. Weingaertner**

King & Spalding
1180 Peachtree Street, N.E.
Atlanta, GA 30309-3521
(404) 572-4826
(404) 572-5134 (fax)
BY: **Bruce W. Baber**

King & Spalding
100 N Tryon Street, Suite 2900
Charlotte, NC 28202
(704) 503-2630
704) 503-2622 (fax)
BY: **Steven T. Snyder**

For Defendant: King & Spalding
401 Congress Avenue, Suite 3200
Austin, TX 78701
(512) 457-2000
(512) 457-2100 (fax)
BY: **Brian C. Banner**

For Defendant: Google
1600 Amphitheatre Parkway
Mountain View, CA 94043
(650) 253-0000
(650) 618-1806 (fax)
BY: **Renny Hwang**

Also Present: **John Mitchell**

THE COURT: Welcome. Please be seated. We are here for claim-construction hearing in *Oracle versus Google*, Case Number 10-3561.

Appearances, please.

MR. JACOBS: Michael Jacobs, of Morrison & Foerster, for Oracle. With me is Marc Peters; Andrew Temkin, in-house at Oracle; Professor John Mitchell, of Stanford University Computer Science Department; and Roman Swoopes, of Morrison & Foerster.

THE COURT: All right. Welcome to all of you. And?

MR. WEINGAERTNER: Good afternoon, your Honor. Scott Weingaertner, of King & Spalding. With me at counsel table are Bruce Baber, of our Atlanta and New York offices; Steve Snyder, of our Charlotte office; Renny Hwang, of Google; and Brian Banner, of our Austin, Texas office, who'll be arguing with me today. He's a fourth-year associate.

THE COURT: Very good. That's great. Wonderful.

Okay. We need to be done by 3:00 p.m., so each side has about 45 minutes. And I'll try to keep track of the time you use. You can put one minute in on one phrase, and 30 minutes on another. You use it the way you want; but when you run out of time, then you sit, mute, for the remainder. So, you know, keep track of your own -- I mean, be mindful of the time that is used.

Now, I do have -- it's okay if you tell me why some

difference between you is important.

In other words, if you say, "This is important because the other side is trying to knock us out on invalidity," I want to know that because, even though we're not supposed to construe the claims with regard to the accused device or the prior art -- so -- or at least -- you know what I mean.

Now, what I have learned on this job is that, for the first five years, the lawyers would talk me into using loaded words, but not tell me why it mattered. And then, later in the trial, I would kick myself, thinking: If I'd only known how they were going to use that loaded word, I would have used a different word. So it's okay with me if you want to say the other side is trying to talk you into this word, because of this piece of prior art they're trying to avoid it, or that they're trying to bring it in, that's fine with me. So I have no problem with that, but I don't want to -- I don't want to -- I mean, it's up to you.

Now, we will come to -- the first phrase is -- I believe it's the "reduced class files," I believe. Is that the phrase: Reduced class file?

MR. JACOBS: Yes, your Honor.

THE COURT: All right. So the plaintiff gets to go first. And you have the floor and I'll try not to interrupt you, unless I have a really good question.

MR. JACOBS: Terrific. And Mr. Swoopes will handle one of the arguments today, your Honor. He's a second-year -- first-year associate.

THE COURT: First year. That's great.

MR. JACOBS: We have a set of slides. I think we'll be jumping around a little bit.

May I approach, your Honor?

THE COURT: You may.

(Whereupon a document was tendered to the Court)

THE COURT: Thank you. You have a set for my law clerk?

MR. JACOBS: I do.

THE CLERK: Thank you.

THE COURT: Great.

MR. JACOBS: So can we go to reduced class file, please?

From the tutorial, I hope your Honor will recall that reduced class files are about this technique of eliminating or moving away or sharing the elements of these class files that are littering memory, and putting them in a shared location. And this was our tutorial slide that we used to illustrate that.

We've got these common elements in Class 1, 2, and 3. We're going to move them into this shared space. And then we're going to refer to the common elements, rather than repeat

them, and unnecessarily use up memory.

The heart of the claim-construction dispute here is whether reduced class files are necessarily class files.

Our proposed definition does not repeat the expression "class file." The Google proposed construction does repeat --

THE COURT: Okay.

MR. JACOBS: -- a class file.

THE COURT: Okay. So what turns on that?

MR. JACOBS: Probably infringement, because the formal definition of a class file may not be repeated in their system.

We're not quite sure about how this is going to play out, because we also have a claim-construction dispute about the meaning of "class file," which we haven't teed up for you today. So there will be some second-order disputes here.

We're focusing today on whether "reduced class file," read in light of the specification, is properly considered a class file.

And Google's argument, at bottom, is: The words "class file" appear in "reduced class file," therefore, a reduced classed file must be a reduced class file, with emphasis on "class file."

Actually, our view is that no formal construction of "reduced class file" is necessary. If you look at the claim

language -- here is Claim 1 of the '702 patent -- it says how you get to a reduced class file. And what you do is you identify the duplicated elements, and then you remove the duplicated elements from these class files, and you obtain reduced class files.

So what is a reduced class file?

Well, it's really the result of going through the steps set forth in the claim. We don't think that any formal construction is necessary; but if there is to be a formal construction, we do not believe that the term "class file" should be in that formal construction.

And the most important reason why that is true is that there is a formal definition of a "class file" in the specification of the '702 Patent. And that formal definition, as we'll see as we walk through the logic here -- you'll see that that formal definition is not met by the description in the preferred embodiment of reduced class files. If we don't embody -- if we don't embrace the preferred embodiment, then the construction can't be right.

So what does the specification tell us about class files?

THE COURT: Say that again; that the preferred embodiment would be read out of the patent if -- if Google's interpretation is correct?

MR. JACOBS: That's accurate. That's right.

THE COURT: Okay.

MR. JACOBS: So we have -- a specification tells us what a class file is. And your Honor had a question at the last hearing here: What are these thing called class files? And the specification tells us that Java classes are compiled into machine-independent bytecode class files. Each class contains code and data in a platform-independent format called a "class file format." So that is the result of compilation of a Java program: You get class files.

The specification also tells us that each class file is self-contained. Now, recall, this is the essence of object-oriented programming, of which Java is a species. When you program in an object-oriented programming language, you keep the data and the code that is associated with it together; and the output is a class file that is self-contained. It has what it needs to do its own internal logic.

That's by contrast to programming mechanisms in which the data is maintained in some data files that are referenced by the executing code.

So we have -- to review, we have a class file format. And a class file is self-contained.

Reduced class files are not self-contained, and so they can not be class files.

How do we know that?

Well, we know that what happens is the duplicated

elements are taken out of what will be the reduced class file, and they're put somewhere else.

These were necessary elements; these constants that are being moved by way of the tutorial into this shared constant pool table. They were thought, if you will, to have been necessary to this class; to this class file. And now they're being moved off.

So, plainly, what we're looking at here at Class 1, under individual classes on this tutorial slide from the '702 Patent -- it's no longer self-contained. Now these shared elements are up in this shared constant pool table. So we have --

THE COURT: Go back to the --

MR. JACOBS: Mm-hm? To the tutorial slides?

THE COURT: -- to the tutorial for a second, and make your point again that you were just making.

MR. JACOBS: So when we created class files through the output of a properly configured Java compiler, we created Class 1, Class 2, Class N. And these were class files because, among other things, they were self-contained. The data in the class file that's needed for that class file is there in that class file, as determined by the original programmer.

Let me try this analogy. I think this works. When the -- when the draftsman of each of these agreements wrote these agreements, he included all of the definitions and all of

the tables at the back of each agreement necessary for that agreement. These class files do not need to cross-reference to some, say, master agreement, where we're going to move the definitions or move essential terms, and then have little pieces, like statements of work, underneath a master agreement.

The rule that Java established is: When you're drafting, it's got to meet the four-corners rule. Everything you need for that class file should be in that class file.

Now, of course -- and this is Google's point -- you may have in that agreement a reference to something external, like an exchange rate, or LIBOR --

(Reporter requests clarification)

MR. JACOBS: L-I-B-O-R.

-- or the CPI, but from the standpoint of whether your contractual draftsman followed the rule, everything needed was within the four corners. There was nothing shared between the agreements, as we see in the creation of reduced class files.

So, by contrast, your four-corners rule got violated when we created reduced class files, because what we did is we moved the definitions and we moved the shared tables at the back of the agreement into a shared area. We put them in a master agreement. And now we're cross-referencing to them. So your four-corners rule got violated.

It's similar. That it is the basic concept in Java.

In object-oriented programming, each class is going to be self-contained. When we create reduced class files, we're going to move the shared data elements into a shared table. Very simply, they're no longer self-contained.

THE COURT: So, because they're not self-contained, you say they can't be a class file?

MR. JACOBS: Correct.

THE COURT: But why, then, are they called "reduced class files"? Why weren't they just called "reduced files"?

MR. JACOBS: It's an artifact of the drafting process.

To jump ahead to my favorite slide of today, Diet Coke is not Coke. The fact that Coke is used there (indicating) does not mean that the formula -- it's -- Diet Coke is not merely Coke with the calories removed. Diet Coke is a different formula to create a similar taste.

THE COURT: I don't know why that's your favorite slide. I don't know. I don't understand the point.

MR. JACOBS: The point is: Using the word "Coke" in "Diet Coke" does not mean that we didn't change the formula of Coca-Cola when we created Diet Coke.

And, similarly, when we created reduced class files, we changed the formula. We changed the recipe. We changed the specification. We no longer conform to the recipe for a class file.

THE COURT: Well, what -- where the -- the brown items that are now in the mclass file.

MR. JACOBS: Yeah.

THE COURT: There must be something in Class 1 that's a pointer, or something that would take the place of the -- so that that would still be in there, wouldn't it?

MR. JACOBS: That's exactly right.

And the important point there -- and this technical analysis is not disputed by Google. They dispute the legal significance of it; but what happens is this -- as is set forth in the specification at column 9, lines 55, a new constant type replaces the duplicated element in the reduced pool, to direct constant resolution to the shared element in the shared constant pool.

So this is -- we've added flavoring to make this work. We've taken some things out. We've added something in.

And here's the real -- the technical clincher to this argument. This new constant type, which is defined with a corresponding constant type tag -- we're looking the column nine, line 55 to 65. We have a new constant type with a new -- what's called an "info[]element." That's just a term in the -- in Java programming. And that new info[]element is an index into the shared constant table.

This new constant type -- if you look at the specification of the Java class file format set forth in the

specification of the patent, there is no constant type that has as its info[]element an index into the shared constant table. This is something that's created not as a result of creating a Java class file, but by doing additional processing on the Java class file, to create what the patent happens to call "reduced class files."

So we process our soft drink to create a diet drink. And, along the way, we do some taking out, and we do some putting in; but it's no longer the original formula.

Google's technical description is not very different from ours. This is their slide from their technical tutorial.

They, too, are showing that these reduced class files have these shared elements moved into a shared constant pool table. And there's got to be a way to reference from those reduced class files to that share table.

That's not part -- our point; not theirs -- our point is: That's not part of the Java class file specification. So again, our point -- our basic argument on reduced class files is that we don't need to construe them. We know how we get them from the claim language.

Class files define a Java class, and are in self-contained class file format; but reduced class files are not in that class file format, and hence, are not class files.

THE COURT: All right. Could I try to say in my own words what I think I hear you saying?

You're saying that if Google's proposed construction were used, and you followed the steps in the specification, the reduced class file would not, in fact, be a class file, and therefore, the patent would cover nothing?

MR. JACOBS: That, I think, is correct. I don't know about the --

THE COURT: I didn't say that very well.

MR. JACOBS: Perfectly.

THE COURT: You're saying that Google -- if you were to do what the patent teaches somebody to do, then it would cover nothing because, according to Google's proposed construction, it would have to result in a class file. And you're -- I think you're saying that if you go through those steps, what you wind up getting is something less than a class file, so the Claim 1 would cover nothing?

MR. JACOBS: That's 99 percent right.

The only thing I would add to it is: You do something less and something more than a class file. You twist it in two directions. You take away things from the class file, and you add things that are not in the class-file format, and therefore, your conclusion is exactly right. It certainly wouldn't cover the preferred embodiment.

THE COURT: All right. Are you done?

MR. JACOBS: Yes.

THE COURT: Okay. We'll hear from the other side.

Now, if you want to do a rebuttal, that's fine. You can do it. It will still come out of your time, though.

MR. WEINGAERTNER: Good afternoon, your Honor.

THE COURT: Good afternoon. Can we give one of these to my law clerk?

MR. WEINGAERTNER: Your Honor, what we find here --

THE COURT: Wait. Wait. We've got to be on the record.

Okay. Now go ahead.

MR. WEINGAERTNER: We think it curious that Oracle has said that it doesn't really want to construe "class file," and then predicates its entire construction of "reduced class file" on an apparent definition of class file; one, by the way, which very much reads in what we think is a specific Java definition of a class file.

We haven't proposed to construe "class file" itself, because it was --

THE COURT: All right. I know you don't, but obviously, it's in the background here. So do you disagree with Mr. Jacobs' definition of this four-corners rule, and so forth, to be a class file?

MR. WEINGAERTNER: Yes, your Honor. I think that it's -- it's a way of defining it that's very strict. And it may be --

THE COURT: Well, why are they doing that? Are they

doing that to avoid prior art? What's really going on here?

MR. WEINGAERTNER: It's a great question, your Honor. What we think --

THE COURT: It's just a question. I don't know how great it is.

MR. WEINGAERTNER: Well, from my perspective, it gets right down to -- what's going on here is that the -- we believe that what Oracle's trying to do is construe this term so broadly, that it will cover virtually anything.

When one looks at the actual language, there's almost no structure to it. A reduced class file contains a subset of the code and data contained in the class file. Technically, that includes zero. A subset includes nothing. It could include the number three. And so -- if it happened to be in a class.

So the idea of this term having any meaning at all -- it simply washes away, and there's just no there there at the end of it.

THE COURT: Well, all right. That's a possible point, but what do you say to the opposite criticism; that you would be -- yours is so narrow, it's impossible to meet, so it would read the preferred embodiment out of the -- out of the patent?

MR. WEINGAERTNER: We don't think that's true. We think that the term "class" isn't as narrowly defined. And,

again, it's difficult, because "class file" wasn't really construed here; but we believe that the term that they're using is an attempt to read in, in some ways, a very narrow definition of a class for purposes of defining class, and then reading it back out again.

And, in fact, we've got an example. And because I'm rebutting, I'm not going to follow linearly through what we had here.

We do believe that the reduced class files satisfy definition of a class file format.

An example is -- and this goes to the issue of what exactly self-containment means. We think that a class can reach out and point to other code or things that the class may need to rely on. In fact, classes have to communicate with each other. And they can't do it if they're fully self-contained in that sense.

And to give you couple of examples, one is -- and this comes, I think, from the Java specification; that there's something called a "LineNumberTable attribute," that is used to point to virtual machine code that corresponds to the original Java source file. So it's an example of something that's in a class file that's actually reaching out for what it needs outside of the class.

THE COURT: Wait. Say that again.

MR. WEINGAERTNER: So, in other words --

THE COURT: So wait. Wait. So you're -- it will help me if -- I know your time is limited, but I do want to -- sometimes it helps if I try to summarize what you've just said.

MR. WEINGAERTNER: Yes, your Honor.

THE COURT: You say Oracle argues that a class file must be self-contained, but that -- that's the four-corners thing that Mr. Jacobs was talking about, right?

MR. WEINGAERTNER: Right.

THE COURT: Everything's got to be within the four corners.

MR. WEINGAERTNER: Right, which -- we have a fundamental disagreement on that.

THE COURT: All right. So you disagree with that.

MR. WEINGAERTNER: Yes, your Honor.

THE COURT: And -- all right. And you say the patent itself has this example where it's not self-contained.

MR. WEINGAERTNER: Right.

THE COURT: And is that your point?

MR. WEINGAERTNER: Yes, your Honor.

THE COURT: All right. So go through the example again, and help me understand why it violates the four-corners rule.

MR. WEINGAERTNER: Okay. So again, the four-corners rule is something that we heard from Mr. Jacobs.

THE COURT: I know. I know. I'm putting that in

quotes, because I'm not saying it's a rule. How would I know? But that's what Mr. Jacobs said.

MR. WEINGAERTNER: Yes, your Honor; just assuming, for the sake of argument.

There is an attribute in this particular Java class definition, 4.7.6, which is called a "LineNumberTable attribute." And it -- again, as it says, it's used by debuggers to determine which part of the Java Virtual Machine array corresponds to a given line number in original Java source file.

THE COURT: You're going so fast, the court reporter will never get that -- okay -- but since it's up there (indicating), I guess she can.

Wait a minute.

-- "debuggers to determine...part of a virtual machine...corresponds to a given line number in the original source file."

So this is outside the four corners? It doesn't say "outside the four corners" there.

MR. WEINGAERTNER: Well, it's basically pointing to the original Java source code file, which is something that's not actually in the class file that's compiled down and running.

THE COURT: Does it say that in that paragraph?

MR. WEINGAERTNER: No. Used to -- again, we didn't

necessarily put this in the tutorial, but it's -- the original source code file is something that's external to the class that's actually being executed.

THE COURT: Okay.

MR. WEINGAERTNER: So that's one example.

And we think when the application was drafted, they weren't drafting it with this specific issue in mind.

And we -- we also think that it's -- we're not sure exactly how it is that someone reads out the terms "class" and "file."

Counsel for Oracle said that this was an artifact of the drafting process. We think that's probably correct. And we believe that, if something other than a class file was intended, it could easily have been labeled as such.

THE COURT: But wouldn't it be -- what's wrong with just saying, "Look. You start with a class file that everybody agrees is a class file. And then you take out something. And what's left over is the reduced class file"? What's wrong with that argument?

MR. WEINGAERTNER: Well, we think it's probably correct, but it still needs to retain something of what it means to be a class file, and not be just anything; because if we look at -- if you really carefully parse what this -- what this term goes to here, as I said before, this definition here could literally be anything. One wouldn't even know what it

would be limited to.

THE COURT: Wait. All right. So who is that talking there?

MR. WEINGAERTNER: Right here? The -- this is the Oracle opening brief.

THE COURT: This is -- Mr. Jacobs' own opening brief says, "Reduced class files do not satisfy the class file format specification."

Well, that's what he was saying a while ago.

MR. WEINGAERTNER: That's right, your Honor; but what I wanted to do was specifically -- oops. Sorry -- was specifically point to parsing this language here (indicating). And when you read it carefully, that's the point that I made earlier, but I would like to reiterate it; that it contains a subset of the code and data contained in the file. So it's something that contains a subset.

We don't know what that thing is. And the subset could be anything. It could actually be the absence of data. So really what they're saying is, it's just any container that may or may not contain anything. So it is devoid of meaning.

THE COURT: Why couldn't we just call it -- it was the original class file, minus what we took out and put in the common area?

MR. WEINGAERTNER: I think we would agree with that, your Honor; but I think that the notion of it being -- of it

retaining a class, and not just being any container with one single data point or nothing -- that's the problem that we have with that.

THE COURT: Well, you're -- you're putting weight on the fact that -- you're emphasizing the word "class." And -- okay.

MR. WEINGAERTNER: No, your Honor.

If I could just maybe clarify, what's unusual about this definition is it uses the term "reduced class file." We should just kind of remove that, and just say it -- whatever it is -- contains a subset of the code and data contained in the class file; but there's no notion of even what that subset is.

So what they're saying is: It could be anything that contains a subset.

And the subset -- as I say, it could be data. It could be data with the number three. And there could be number three stored in any container of any kind -- would now suddenly kind of fall within this -- this construction. Kind of -- collapses down to something with no meaning whatever.

THE COURT: And what if we didn't go with anybody's construction, except my own, and said, "It's the original class file, minus whatever you took out."

MR. WEINGAERTNER: Your Honor, I think --

THE COURT: In other words, the reduced class file -- there's the class file you start with, and then there's the

reduced class file, which is what's left over after you take out the common -- common code.

MR. WEINGAERTNER: On the fly, your Honor, I think that's probably something that we could live with, although I'd like to be able to confer, if I could, about that; but I think it gets --

THE COURT: That wouldn't necessarily mean it would mean -- what's reduced wouldn't necessarily be a class, file in the sense that it would meet all of the four-corners thing. It would still have to go borrow the original -- the data that was taken out and put in the common area.

Well, okay. I'm taking up your time here, but you continue on.

MR. WEINGAERTNER: Well, actually, I think that really kind of got to the nub of what we were trying to argue, your Honor; that we think it's improper. We think it's kind of really cardinal error of claim construction to read out the words in the claim term, itself. We don't find any authority to support doing that.

We're not sure we understand the Coca-Cola analogy, although my partner, Mr. Baber, is Coca-Cola's lead trademark counsel and IP counsel, and he might be able to shed some light on whether that --

THE COURT: I'll give you an example, since I'm wasting everyone's time. Let's say you had a patent that

starts off with a red car. It's got a red car. And the patent deals with recoloring cars. So the phrase was, "the recolored red car." So what it's saying is: Yeah, it used to be red, but now it's been recolored, so it's not going to be red anymore. It's going to be green, or whatever; but recolored red car. It's kind of an awkward thing, but maybe the -- this phrase is a bit awkward as well.

And so that the key word is the "reduced" word, rather than -- and "class file" is just -- "class file" is just like one word.

I don't know. You know, it's frustrating that these gigantic law firms can't write a clearer language, isn't it? That --

MR. WEINGAERTNER: Yes, your Honor.

THE COURT: -- that we have to go through this process instead of --

All right. Anything more?

MR. WEINGAERTNER: No, your Honor. Thank you. That summarizes.

THE COURT: All right. Let's hear the rebuttal on -- rebuttal.

So would you agree with that -- what -- my suggestion, which would be what's left over, after you take out the -- in other words, the class file minus the reductions?

MR. JACOBS: I think so. I think, just to reinforce

it, that is what the spec -- what the claim language tells us.

So again, we start with class files. They have duplicated elements. We form a shared table comprising a plurality of the duplicated elements. We remove the duplicated elements from the plurality of class files to obtain a plurality of what the patent calls "reduced class files."So reduced class files are the result of removing the duplicated elements.

The important point is that it's not necessarily a formal class file.

This reduced class file is a term --

THE COURT: Why don't you all -- or did you? Did you brief what the meaning is of the word "class"? Why did you punt that to -- kick that can down the road? Maybe that's important to deal with right now.

MR. JACOBS: It could be, your Honor.

And I think one of the things that has to become apparent as we have evolved towards this hearing is the importance of making you aware of where there are consequences to decisions now that -- I want to avoid surprises to the Court. And we do have differing views on "class file." And just as, again, the process of getting down to six terms unfolded, we didn't end up choosing that term to debate.

THE COURT: Mm-hm. All right. I did put limits. Okay.

MR. WEINGAERTNER: Your Honor, in the interest of possibly resolving this right now, we think we could live with the possible proposed definition your Honor mentioned, as long as it retains some notion of being a file.

The way we read construction --

THE COURT: Well, the word the "number three" could be a file. I know that much. You could have one little data point in a tiny little file, but it's still a file.

MR. WEINGAERTNER: We would agree that, as long as there's a file containing that, and not just the data itself, we would agree with that, your Honor.

THE COURT: Well, but no. A file can cane data only, right?

MR. WEINGAERTNER: Right, but the data itself would need to be presented as a file, and not just pure data, free of any container.

THE COURT: Well, I'm not sure I understand the distinction you're making, but you're -- Mr. Jacobs has the floor now. So --

MR. WEINGAERTNER: Yes, your Honor.

THE COURT: -- go ahead, Mr. Jacobs.

MR. JACOBS: Your Honor, I think it might be helpful if -- I want to try and help the Court get this right. And it may be that we should, following this hearing, take another shot of trying to work through this definition with the aid of

the Court's guidance.

Let's make sure we know what we mean by "file," so we don't have a dispute about the construction. That's the only -- that's the concern I have with Mr. Weingaertner's suggestion that file be intruded.

THE COURT: All right. Three more minutes. Okay. Let's go to the next phrase, which is -- which is -- it's in the '520 Patent.

MR. JACOBS: Play execution.

THE COURT: I think that's -- "the play executing step."

MR. JACOBS: Mr. Swoopes is up, your Honor.

THE COURT: All right. What's your name again?

MR. SWOOPES: My name is Roman Swoopes, your Honor.

THE COURT: How do you say that last name again?

MR. SWOOPES: Swoopes. S-w-o-o-p, as in "Peter," "e," as in "Edward," "s," as in "Sam."

THE COURT: Thank you. Please proceed.

MR. SWOOPES: Thank you, your Honor.

The issue before the Court today is whether a person having ordinary skill would be able to determine the meaning of "the play executing step" as recited in Claims 3 and 4 of this patent, or whether, as Google contends, the term is indefinite, and cannot be construed.

Claims 3 and 4 in this patent depend from independent

Claim Number 1. And Claim 1, as you can see, contains five steps. It's a method for statically initializing an array that first involves compiling source code; receiving the class file that is generated from the first step into a preloader; simulating execution of the bytecodes in the class file; storing the output of an instruction requesting the static initialization of an array; and interpreting that instruction.

As the Court can also see, Claim 3 of the patent further refines one of the steps above, and says, "where the play executing step includes further substeps allocating a stack and performing manipulations on the stack." So, from Claim 1, we see that we basically have a one-in-five choice, even without seeing or knowing anything else.

And from the language "receiving" -- the play executing step doesn't sound like receiving. Excuse me. From "compiling," it doesn't look like compiling, either. There's not an overlap of work. "Simulating execution" is about the closest that one would get, without looking at anything further. And, similarly, "storing and interpreting" don't seem to, on their face, be related to play execution.

If we briefly look at the language of Claims 4 and 5, they follow the same basic pattern as Claim 3. They recite refinements of the play executing step. Claim 5 is not asserted in this case, however, your Honor.

Now, even without knowing anything more about the

patent, we would have a one-in-five chance of getting this right, but we see that this --

THE COURT: What is this? Russian roulette? This is a new claim construction rule: Just flip the coin?

MR. SWOOPES: Well, I make that point just simply to say, your Honor, that a person having ordinary skill has some choices in front of him or her, without having to think much about it; but in fact, it's --

THE COURT: All right. Now I see your point.

MR. SWOOPES: Yes.

THE COURT: So it's got to be one of those five things.

MR. SWOOPES: Yes, your Honor.

THE COURT: All right. So we've got to look at each one and say, "Is that one?"

MR. SWOOPES: Right.

THE COURT: "No, it's not that one." And then we go to the next one. And pretty soon, it will be down to one or two?

MR. SWOOPES: Yes, your Honor, exactly; but we don't even have to do that, because the specification makes it quite clear for us. It explicitly defines that "play executes" means "simulate executing."

About the only thing that the patent could do to make it clearer would be to say, "By 'play executing,' I mean

'simulates executing.'"

And we see that the specification does this in numerous places. We simply have illustrated here the summary of the invention, but if one is to simply substitute "simulates executing" into the location here of "the play executing step," it would not change the meaning of the claim at all; the method of Claim 1, wherein the simulating -- the simulates executing step includes further substeps. It's a straightforward transformation, your Honor. It's explicitly defined within the specification.

I don't want to take up too much of the Court's time with this straightforward substitution, so I will -- I'm happy to answer the Court's questions.

THE COURT: All right. I want to save your time.

MR. SWOOPES: Good.

THE COURT: Good. Thank you.

MR. SWOOPES: Thank you.

THE COURT: All right. Who's next?

MR. BANNER: Thank you, your Honor. Brian Banner, on behalf of Google.

THE COURT: Okay.

MR. BANNER: Your Honor, I'm going to be addressing "the play executing step" today. It's a term, as Mr. Swoopes has said, that is in Claims 3 and 4 of the '520 Patent. And I am going to discuss why we believe that this claim lacks proper

antecedent basis, and why that renders it indefinite in this case.

THE COURT: Well, how can you say that? It's got to be one of those five steps, doesn't it?

MR. BANNER: Well, your Honor, our position is that that's kind of a superficial way to look at this.

THE COURT: Whenever a lawyer says, "Our position is," I know that they have some doubt about, because they -- it's just their position. It's a funny thing, but the phrase "Our position is" -- just as a hint for the future -- is a signal that you are not sure of yourself.

MR. BANNER: Thank you, your Honor. I will --

THE COURT: Keep that in mind.

MR. BANNER: I'll put that in my back pocket.

THE COURT: All right. That's your position, but --

MR. BANNER: Okay.

THE COURT: -- but there's only five steps it could be. It calls out five steps. It's got to be one of those five.

MR. BANNER: Your Honor, we do not dispute that when this claim was originally --

THE COURT: The one in the middle seems pretty logical, doesn't it?

MR. BANNER: When this claim was originally written, and if you ignore the prosecution history, which is also part

of the intrinsic evidence, I would agree with you.

THE COURT: Yes.

MR. BANNER: But I would think that today, based on the absence of Oracle's discussion of the prosecution history, that should be a big tell as to what the deficiency in their arguments --

THE COURT: Is there something in the prosecution history that blows that out of the water?

MR. BANNER: Yes, your Honor, there is.

THE COURT: All right. Let's see that.

MR. BANNER: Claim 1, as originally filed, was filed as stating "play executing the bytecodes," which is --

And then Claims 3 and 4, as we said, still -- when they were originally filed, still also recited, you know, "the play executing step." However, Claim 1 was rejected by the Examiner over prior art. Claims 1 and 3 both were.

The parties got together. And there's a little more detailed discussion of the actual prosecution history in our opening brief, but this is the gist of it. They had an Examiner interview. And they agreed that this amendment would narrow the claims. And what they did, as shown here, is they added "simulating execution of." They deleted "play executing." And then they added, "without executing the bytecodes" as well. And so during prosecution, they changed the scope of this claim and they changed the language of

Claim 1.

Claims 3 and 4 were not similarly changed, which brings us to --

THE COURT: How did 3 and 4 read before the amendment?

MR. BANNER: Claims 3 and 4 read as Mr. Swoopes showed you in his slide. They still read "the play executing step" of Claim 1. I don't have a slide showing that.

THE COURT: Well, so at that time, it was referring to what you got there in brackets?

MR. BANNER: That's correct, your Honor.

THE COURT: So it was clear then it was referring to that language?

MR. BANNER: That is correct, your Honor. It was referring to that step at that time.

THE COURT: So then why doesn't it still refer to it?

MR. BANNER: Well, the problem is that the -- the reference to "play executing" -- not only is there -- it lacks antecedent basis, which we could just say -- well, that's just a simulating-execution step, but this was clearly a narrowing amendment. The Examiner said here in the Examiner interview that this narrowed the claims. And so because it's a narrowing of the claims, we submit that "play executing" and "simulating execution" are a not synonymous, as Oracle has argued to you today.

According to the --

THE COURT: A little early. Okay.

MR. BANNER: In the Examiner's mind, the Examiner required them to change "play executing" to "simulating execution," without executing the bytecodes. And that evidences the Examiner's belief that these two terms are not coextensive.

And so Oracle would like you to believe that these are just synonymous. We can just point back to the other one, but we submit that that is not the case.

THE COURT: But before the amendment, didn't it clearly refer to that limitation?

MR. BANNER: I would agree, your Honor, it did.

THE COURT: So isn't it more likely that it was just a screw-up, and they should have changed 3 and 4, but didn't do it?

MR. BANNER: Well, your Honor, that's one way to look at it, but there's also another problem with this -- with accepting that explanation. And that's the Oracle's proposed construction would still be indefinite.

This idea of, hey, it's got to be one of these -- you know, one of these steps; we're just going to point to the one that looks, you know, the most reasonable -- that sounds good, but it's really just a superficial problem that you're fixing there. And it doesn't fix more underlying indefiniteness

problems with Claims 3 and 4.

And, as we show here, they didn't amend Claims 3 and 4, but they did amend Claim 1. And part of that amendment was this additional limitation that -- they say, "Claim 1, you simulate execution of the bytecodes without executing the bytecodes." So by that amendment, we when they narrowed it, they said, "We're no longer executing the bytecodes as part of this claim."

Well, the problem becomes, when you now look at Claims 3 and 4, you're reading a bytecode, and then you're actually performing the stack manipulation. You're performing these operations that the bytecode tells you to do. And so there's a contradiction here between Claim 1, and Claims 3 and 4 that -- it just -- we cannot square, your Honor. This -- this language does not make sense.

THE COURT: Well, okay. Possibly.

I think I see that, but if the Court were to construe "play executing step" to refer to that metal step -- the one about simulating -- would you still have your indefiniteness argument based on this language, without executing the bytecodes?

MR. BANNER: Yes, we would still have that argument, your Honor. And that's why we believe that this entire operation of trying to correct these two claims, out of the 132 that have been asserted in this case, is -- is -- is a

rather -- we're not exactly sure why Oracle's chosen to do this here, because they still have these other problems. And their claim construction doesn't fix the problem. All it does is it highlights the fact that these types of problems -- these type of claim-drafting errors, as we briefed in our opening brief -- these should be fixed by the PTO, and they should not be dealt with in this setting, where it's unclear what the patentees were trying to get at when they amended these claims.

THE COURT: All right. Thank you.

Any rebuttal?

MR. SWOOPES: One brief rebuttal statement, your Honor. And that is simply to say that simulating these instructions in an artificial environment does not equate to actually executing these instructions.

THE COURT: Thank you. We will now go to the next phrase, which is the '104 Patent. "Intermediate form code" and "intermediate form object code."

MR. JACOBS: So the '104 Patent, your Honor, is the resolution patent in which we -- this is the one that dates back to 1992. And it's the patent in which a symbolic reference is resolved into a numeric reference by the claim language, as illustrated in our tutorial. The process here is taking an instruction of the form LOAD *Y*, where *Y* is the symbolic reference, and converting it into the location in memory, referred to here as Slot 2, and doing that once, so

that when you go through that step again, you don't have to resolve the reference.

So my analogy for this is that if one is approaching my house at 36 Bulkley, one would be looking down the street, and looking into the house address, and trying to find it. The next time one comes to that house, it's a lot easier to remember third house on the left, and not have to look down at the what is, in effect, an address by name, rather than a location on the street. So that's the way this works. The resolved reference is relied on for subsequent execution.

The dispute between the two -- between the parties is actually narrow by words, but important in its substance. We have mostly aligned proposed construction for intermediate form object code. The difference is in the word "executable." And we propose that the that intermediate form object code or its variants in this patent must be construed as executable, because the specification tells us so.

And, in particular, in the summary of the invention, it states right up front that what we're talking about here is generating executable code, and resolving data references in the generated code. So the very first line of the summary of the invention tells us that we're talking about executable code.

THE COURT: All right. So go back to that one again. I want to --

MR. JACOBS: Mm-hm.

THE COURT: All right. Thank you.

MR. JACOBS: And then further in the specification, we are told that, under the present invention, the compiled intermediate form object code -- so that's that code that was generated by the compiler -- will achieve execution performance. It is, therefore, executable.

And if we put these two back to back, we're going to generate executable code. And that intermediate form object code is going to achieve execution performance. Plainly, what we're talking about is that, in conjunction with intermediate form code or intermediate form object code, is executable code.

And when we look to the claims, we see that the code must be executable as well. In Claim 22, for example, we see an apparatus containing a compiled program in intermediate form object code, and then a process configured to execute the instructions in that program. So plainly, that program is in executable format.

Now, what's going on here is that the word "intermediate" is used twice in the specification. And there is, in compiler architecture, this concept of an intermediate representation, as illustrated here in Figure 4 of the Patent. The intermediate representation is the representation of your original program in some internal form that the compiler is going to work on along the way to creating and generating what

is referred to here in Figure 5 as "intermediate form object code"; but as illustrated here, plainly, the intermediate form object code is the output of this process. And, as discussed earlier, that intermediate form object code is executable.

So this is all about Google's attempt to read some prior art about intermediate representations; about what went on in the compiler prior art; the kind of processing and optimization that was done by way of internal compiler architectures on the very distinctive architecture -- the really highly innovative architecture -- of this patent, which is about the Java architecture.

And to just bring it home, what we're talking about here is executable code in the Java context; Java bytecode that executes on a Java Virtual Machine. The Java Virtual Machine is the execution platform. Within the meaning of that claim language, the Java Virtual Machine configures this process to execute the code that we submit, by definition, is executable.

And similarly, in Android, we have a virtual machine configuring the processor. The Java bytecode that's been translated to .dex code is in an executable format.

We are not talking about -- when we talk about "intermediate form object code," I'm pointing to the middle of the Java compiler box, or the Java compiler box in Java and Android. We're not talking about what happens internally to the compiler. We're talking about the output of the compiler.

So we both agree that intermediate form object code needs a construction. This is not a case where we are, as Google would have it, reading a limitation into the claim language. We're applying the proper construction to the phrase "intermediate form object code."

And the specification is quite clear, quite explicit, that what we're generating here is executable code. Intermediate form object code is to be executed, and achieve higher execution performance. And the claims themselves, when you walk through the claim language, require that the intermediate form object code be executable.

THE COURT: Is this an invalidity or an infringement issue?

MR. JACOBS: I think it's an -- because they would take the word out of our proposed definition, it's a clue that we're talking about invalidity. And, as we look at their prior art that they've provided to us, I think 90 percent of it is about compiler -- what happens inside the compiler at that intermediate representation stage.

THE COURT: All right. Let's hear from the other side.

MR. WEINGAERTNER: Yes, your Honor. We are absolutely in agreement that the only issue is whether or not the term "executable" should be read into this limitation that's up for construction. And it is an invalidity argument.

We should mention that all of the claims in this patent and, in fact, all of the claims in all of the patents have recently been taken up by the Patent Office. They've granted reëxamining requests on all of the claims of all of the patents-in-suit in this case, including this one, over the art that counsel for Oracle refers to.

THE COURT: When did that happen?

MR. WEINGAERTNER: It happened as recently as yesterday in one of them, your Honor. And in the last couple of days. It's happened over the last few weeks.

THE COURT: All right.

MR. WEINGAERTNER: And so somebody has concluded that the compiler art that's referred to is at least raising a question of patentability in these claims.

We believe that this is a very clear attempt to read a limitation into a term that doesn't have it in there; an attempt to read it from the preferred embodiment. It happens sometimes. And sometimes there's an explicit definition, and sometimes there's a reason to do that; but we don't think that's the case here. We think it's clearly a very deliberate attempt specifically to avoid the prior art.

And what was interesting that wasn't mentioned is that the executable modifier had previously been pending in the -- previously been in the application that led to the patent-in-suit, which was then modified by our issue, which

broadened the claim, and eliminated that modifier.

And so we think it's telling -- and it's not something that Oracle can run away from here. And I'd actually come up with a metaphor that I wanted to talk about, which involve the car that's red. And I wanted to just go through that for a moment.

In other words, if you claim a car in the preamble of a claim, and then later characterize that car as being red, that's fine. The car can be red; but that doesn't mean that car, in itself, is red.

And that's, we think, what Oracle is trying to do here. They're trying to read back in a limitation that's a feature they decided -- referred to in claims as a separate limitation referring back to the general concept of intermediate form code; but that, in fact -- if anything, that, we think, strengthens our view that it needn't intrinsically have that property. Otherwise, you wouldn't need to add it.

Yes, your Honor.

THE COURT: You said something I want to --

MR. WEINGAERTNER: Yes, your Honor.

THE COURT: -- right in the middle. It was explicitly deleted as a modifier. Go through the details of that argument.

MR. WEINGAERTNER: Yes, your Honor.

So Claim 1 of the original '685 Patent, which was

then actually the subject of, we think, two reissue patents, and then a continuation of a reissue -- so what we're dealing with here is sort of the second or third coming of what was originally filed. And that was done to broaden the patents. And a lot of interesting things happened which will be the subject of other things I'll raise before the Court at an appropriate time; but when you look at Claim 1 of the original patent, executable code was clearly recited, per se.

THE COURT: You mean of the application, or of a --

MR. WEINGAERTNER: The issued patent, your Honor.

THE COURT: Oh. It was issued?

MR. WEINGAERTNER: Yes, your Honor.

And then it was -- they attempted to correct and sought correction through a first reissue and then a second reissue, and then filed yet another application off the second reissue. This is years, now, into the life of the patent. I think it was maybe issued for five or six years when this change occurred.

And there was -- obviously, a lot of thought had gone into this. And one of the things that was apparently recognized as being broadening in some way was to eliminate this term, "executable" -- and in the notion of what an intermediate form object code would be.

We think it's telling. And we think that suggests that they needn't be the same thing; that these don't fuse

together automatically and permanently for all --

THE COURT: I don't -- that's '685 versus '104, but is the '104 the same patent?

MR. WEINGAERTNER: That's a -- your Honor, so what happened was -- and I apologize that we don't have a map here to kind of guide your Honor through it, although I think we do have an Elmo, and we may have the patent that I could put up there for your Honor; but just in brief, I'll try to do it from memory.

So the '685 Patent was filed December of 1992.

I think in '96, then-Sun decided to seek broader claim coverage in certain ways. And they tried to peel apart one part of the claim from another part of the claim. And in doing that, they did various things to what the claims constituted.

The '685 Patent was originally surrendered, so that's not in force anymore. And that's required when you seek a reissue.

And they filed a second reissue, broadening reissue; and then a continuation, which ultimately led to the '104 Patent which is in suit today.

And so what we're doing, simply, is contrasting the issued claim in the '685 Patent, which is the lead claim. And the Claim 11 here is the first asserted claim, because the other ones are no longer in the patent.

THE COURT: But what -- is Number 11 the analog to Number 1?

MR. WEINGAERTNER: That's the way we view it. It's basically the lead-off claim.

You know, an applicant for a patent is free to do what they want. And they could actually have broader claims down further in the chain; but it does track, in our view, the -- you know, what was -- it intended to be capturing what the alleged invention is.

And so the fact that executable was dropped, even if it occurs later in the claim, which it does, that gets back to the -- the metaphor of the car that is then, you know, determined to be a red car.

In other words, if you -- in the preamble of a claim, you refer to -- you know, earlier on in the claim you referred to it first instance of the thing that's not so limited, and then later you limit it, you're certainly free to do that, but if anything, that suggests that the first recitation doesn't require that limitation.

THE COURT: So -- but the language doesn't seem to be otherwise the same.

In other words, usually when this argument gets made, all of the words are exactly the same, and the word "executable" is no longer there; but the wording is totality revised. On --

MR. WEINGAERTNER: Absolutely.

THE COURT: There's no -- where -- so how do we --

MR. WEINGAERTNER: We absolutely agree, your Honor. And -- because there is enormous flexibility and freedom in how people can present claims. And, in fact, that's really an issue in this case.

We have a disclosure that's very limited. And now we have 30-some-odd claims, all turning or independent -- I'm sorry. There are a lot of claims that all turn on different usages of the word "resolve"; but here, I think the thing we're doing is focusing in on just the use of this term itself.

And obviously here, there was an attempt to equate intermediate form code as being executable. And later on in the broadening reissue, because it was a broadening reissue that they trace back to the allegation to file within two years of the original issue date of the patent, although we dispute the entitlement to do that, they could have retained the executable notion with the intermediate form code object as -- when they introduced it into the claim, and elected not to. We think that's evidence that the term doesn't intrinsically require being executable.

THE COURT: So you say object code does not have to be executable.

MR. WEINGAERTNER: That's correct, your Honor.

And we -- but this, perhaps, is a time to transition

to point that Oracle made about this notion that -- of intermediate representations.

Oracle did not put anything about intermediate representations in their definition, and sort of introduced this later as -- you know, as a different approach to try to read in a negative limitation.

In other words, what Oracle is trying to do is not only trying to read in a term -- executable -- but it's trying to read out, without actually putting it in the proposed construction, this intermediate form.

And we don't believe that that's -- well, first of all, we don't believe it's proper to try to construe a term without literally putting it into the language that the jury will see, because the jury will be left, I suppose to, to sift through the briefs to figure out what exactly this is going to mean. We think that is a bit of a red herring. And we think it's just a simple attempt to -- again, to read a limitation in from the claim that isn't required. It's not defined. And again, we believe the prosecution history makes fairly clear that it needn't be there.

THE COURT: Okay.

MR. WEINGAERTNER: That's our position.

THE COURT: Thank you, your Honor.

MR. WEINGAERTNER: Thank you, your Honor.

THE COURT: All right. Go ahead.

MR. JACOBS: The problem with Google's definition is that it does nothing with object code. And object code is understood to be executable. And that's why we included executable code in our proposed definition.

In Google's definition, object code is not -- the object in object code is missing. The prosecution history is much more complicated than to show -- than as, really, Google's counsel explained.

What the -- what the applicant did -- what Gosling did in the reissues was separate out what was happening on the compiler side from what was happening on the -- if you will, the user computer side.

You recall the first claim was a system for generating, and then for executing. And if you look at -- as we explained in the brief, what happens during the reissue is he realizes: Wait a minute. I better claim separately what's happening on the server side of things and what's happening on the user computer.

So the word gets changes to "intermediate form object code"; but we still have to construe intermediate form object code. And the best fit -- the definition that comports with the specification -- is executable code. Google's construction doesn't explain what happened to the object in object code.

THE COURT: All right. Let's go to symbolic reference.

MR. JACOBS: We will be saying in a couple of places, your Honor, that the claim language is sufficiently clear that construction adds unneeded complexity; but if construction must be had, we have a proposed construction.

And that is true with "symbolic reference." We think that a symbolic reference is clear in the claim, and that it doesn't need to be construed; and that Google's proposed construction adds a whole lot of confusion and unnecessary complexity. So symbolic reference is in the claims by -- in relief against numeric reference.

And so what we see in Claim 17, for example, is that we're going to resolve the symbolic references to a numeric reference. And, although we haven't teed it up to you the construction of numeric reference, we know that a numeric reference is a location in memory.

Recall Slot 2. So Slot 2 is the numeric reference in the illustration. That's a location in that table. It's a location in memory. The *Y* is the symbolic -- is the symbolic reference there. So once you know that you're resolving a symbolic reference to a location in memory, the symbolic reference is a reference by symbol; but if we have to construe it, then let's look at what the specification tells us.

And what the specification tells us is that the symbolic reference is in the form of a name. So in the illustration that we were looking at in Figure 1, the LOAD

instruction references the variable *Y* by the symbolic name *"Y."* So a symbol is a name.

Google relied on a dictionary: The Microsoft Dictionary. And this, too, helps us understand the relationship between a symbolic reference and a numeric reference.

And, in particular, if you look at the definition of "symbolic address," not perfectly aligned with "symbolic reference," but I think the definition helps us understand what's going on.

A symbolic address is a memory address that can be referred to in a program by name, rather than by number.

Now, what kind of number are we talking about? Because, plainly, a symbolic reference can include a number. It could be *Y*2 or *Y*3.

Well, this rest of the definition helps us understand that.

"The interpreter compiler or assembler" --

In our invention here, it's the interpreter.

-- "translates the name into the number that specifies the address."

So symbolic reference into numeric reference; and reference by name into a reference by location in memory.

Google proposes to add "dynamic," because the specification -- the authority for that is that the specification says the data reference is static -- i.e., numeric -- or dynamic -- i.e., symbolic.

And so in interpreting "symbolic reference," they propose to add in the word "dynamic"; but the specification chose -- the patentee chose the word "symbolic" for the claims. And to say that dynamic equals symbolic, therefore symbolic should mean dynamic doesn't help us at all. What does "dynamic" mean?

And what -- we end up in this kind of a circular loop, where, if we have "dynamic," well, what does "dynamic" means? It means "symbolic." So "symbolic reference" means "symbolic reference," which just got us back to where we started from.

That's probably the proper role of "dynamic," but "dynamic" is one of those words in the computer arts that has infinite meanings. It's a highly elastic term.

Google also proposes to add string- or character-based.

Well, what is that all about? Is it designed to exclude some kind of a symbolic reference, where the symbolic reference is a reference by name?

We're not really sure where they're going with this, but the jury's going to have to be taught what a string- or

character-based reference is, if Google's construction is to be adopted.

THE COURT: I'm -- this is confusing me. I'm sorry.

MR. JACOBS: Okay.

THE COURT: Give me, in very simple terms, where you two disagree. I've lost that point.

MR. JACOBS: We disagree, in that we propose that a symbolic reference is a reference by name, if we're going to construe the term.

They propose that "symbolic" is a dynamic reference to data that is string based or character based.

THE COURT: What's the difference between that?

MR. JACOBS: Well, what does "dynamic" mean?

In the specification, it says that it's symbolic. So that doesn't help us, because we were construing "symbolic reference" --

THE COURT: Give me a concrete example where it would fit yours, but not theirs, and vice versa.

MR. JACOBS: I don't think we know yet. I don't think we know exactly where they're going, honestly.

THE COURT: Where are you going? Give me an example that you think fits your definition.

MR. JACOBS: Here's where we're going: The Android code.

So the Android code has a routine. It's the class.c

routine. And it says, "We're going to link and resolve." And it says, "This converts symbolic references into pointers."

So this -- we think that what is -- is a symbolic reference here is clear to the ordinary programmer; the ordinary person in the art.

Where they're going with their definition to say, "This symbolic reference is not a symbolic reference," we're not sure. We don't have the full ventilation of their noninfringement contention here. It must have something to do with "dynamic" and "string based" or "character based," but since the word "symbolic reference" is used right in the code, we don't think there's any trouble by saying, "A symbolic reference is a symbolic reference," or, at worst, if one has to construe it, it's a reference by name, which is what the specification tells us. So we think it's clear. The claim scope here is clear, without a construction of symbolic reference.

THE COURT: You don't want a definition, because you've got -- is that their own document?

MR. JACOBS: It is. It's the Android code, your Honor.

THE COURT: It uses the claim language right there. So you're happy with that.

MR. JACOBS: Very.

THE COURT: But they -- they're not happy with that,

and they would like to have it larded up with some other definitions.

MR. JACOBS: Like "dynamic," and "string-," and "character based."

THE COURT: Well, I don't know the answer to this, but you know, each of you are saying that certain -- why don't you have a tradeoff? Why don't you each agree that we won't give a definition; you'll let them just fight it out in front of the jury; and each of you give one where there's no construction.

MR. JACOBS: I think that's a terrific idea.

THE COURT: I had a case once where the lawyers agreed that we would construe nothing. It would just be fought out like the O.K. Corral front of the jury. And it was great. They both wanted it that way.

MR. JACOBS: And the sentiment underlying this is that this claim-construction process -- and I think this case illustrates it -- has, to some degree, taken on a life of its own.

If we have code that says "symbolic reference," and we have a claim that says "symbolic reference," why are we construing some --

THE COURT: That's an excellent jury argument, but it could be -- there have been cases where, you know, words don't mean what they say.

MR. JACOBS: But then you should have a very clear signal that we're doing something to narrow "symbolic reference." And we don't see that in the specification.

THE COURT: All right. Okay. All right.

What do you say to that?

MR. WEINGAERTNER: Well, your Honor, we believe that the proposed definition that we have --

And I've got my pointer here.

-- actually provides some guidance for the jury that they wouldn't otherwise have, and does so in a way that's very consistent with the specification. We think the fact that the parties differ to some extent suggests that they really should be construed, although I'm sure we would have no problem battling it out, as your Honor suggested, in court.

THE COURT: But how would you get around the fact that your own document says "symbolic reference"? That's the code; the claim language. So then you would have to say, "Well, yeah, that's what -- they do say the same thing, but 'symbolic reference' in the patent means something different."

MR. WEINGAERTNER: Well, your Honor, I guess just a couple of quick comments on that. We got an e-mail yesterday from counsel saying that they intended to introduce this today.

We hadn't heard that from them earlier.

THE COURT: Introduce what?

MR. WEINGAERTNER: The code that they flashed up on

the screen --

And, just as a matter of mechanics, when we were last before your Honor two weeks ago, we'd agreed that we would supplement our noninfringement contentions next week. And so in some ways, we think it's maybe a little bit premature to refer to "comments."

That's not actual code, by the way. That's comments from the inventor that --

THE COURT: Yeah. Comments. Right.

MR. WEINGAERTNER: -- may or may not be probative, and that may or may not be relevant here. And so our point is that a juror might not have a sense of what this actually means.

We don't believe that string- or character based is difficult for a jury to comprehend. We think it provides useful guidance. And that's consistent with the specification, and would help them apply that language.

We think that Oracle actually concedes that the specification defines what it is to be symbolic. And we think that their definition would read that limitation out, and somehow broaden it, and perhaps be confusing, because there's something actually quite subtle here about the distinction between "symbolic" and "numeric."

I was actually going to use an example, your Honor, if I might. And there was several kind of commonplace examples

that Oracle uses in it papers.

I think you could think of the distinction between symbolic and numeric as imagining baseball players, for example, in the Giants, who have a number on their back. The number on their back has some identity for maybe who they are, but it has no role on where they actually appear in the field. So Tim Lincecum has "55" on his back, but he actually goes, at some point, to Fielding Position Number 1. And Buster Posey has "28" on his back, and goes to Position Number 2, unless the manager decides maybe he should go somewhere else.

And so the number on their back would be symbolic in some way of who they are, but it doesn't ultimately fix where they end up on the field, which would be kind of a rough metaphor for the memory. And we think that it's difficult for a jury to tweeze those things apart, and that it's the string nature and the dynamic nature -- "dynamic" meaning you don't know where that person's going to go; you don't know where that data's going to end up. It's not static.

And that's basically -- we're actually referring to the same text that Oracle did that makes very clear that the numeric and static -- and "i.e." is actually a direct equation. I'm not a Latin scholar, but I think it means "that is." So I don't think there's any question about whether they're basically interchangeable. And the same goes for "dynamic."

You know, I guess we're hoping that the jury will

understand the "dynamic" means, you know, in motion in some way, and not fixed or static. We believe those terms are relatively well understood.

THE COURT: Well, just looking at that, static -- "main interpretation determines if the data reference is static; i.e., numeric." So that -- that seems to be equating "static" with "numeric"; like, say, the number 7.

And then it says "or" -- which I read there to mean "versus -- "dynamic equates to symbolic"; like in *Y*, or *Y*2. You could have *Y*2?

MR. WEINGAERTNER: Yes, your Honor. Yes, your Honor.

THE COURT: All right. *X*1. And -- but that would be -- if you just had one, that would normally be thought of as numeric; but if you have *X*1, that would be symbolic, right?

MR. WEINGAERTNER: That's right, your Honor. And so that's why, down here, we give an example of, you know, Object 3 contains -- it's a string. It's human-readable. It contains a number, for sure; but it's not numeric in the narrow sense that's required here in this field. This field uses these terms differently, we think, than it is necessarily used in other contexts.

This notion of static and numeric -- it's not completely obvious to somebody who's not familiar with this. And because numeric has more to do with an ultimate location -- and there's actually, you know, some disclosure in the patent

that is helpful there.

So in the description here -- and this is looking back at the prior art, because this is, again, not anything that was specific to Sun when they filed their application. The numeric reference here -- as you can see, 2 is -- actually, it's not merely a string. It's not merely a label. It's not merely a number on a ball player's back. It's actually where -- it relates to where something will actually reside that's known; whereas symbolic reference here, although it's very difficult to tell from this figure because it's not dynamic enough, in a way, this really could wind up being anywhere. It's not a specific known slot. "Slot" here actually means this slot, per se.

And so what we've tried to do is to create some -- a tool kit for the jurors to somehow tweeze apart this pretty subtle distinction. And we think that reference by name doesn't really get the jury there.

THE COURT: Go back and show me the claim language that's at issue here.

MR. WEINGAERTNER: (Indicating)

THE COURT: No. I mean --

MR. WEINGAERTNER: I'm sorry.

THE COURT: That's just too -- of course, I don't know. In context.

MR. WEINGAERTNER: I'm sorry, your Honor. I'm not --

THE COURT: The claim itself.

MR. WEINGAERTNER: Okay.

THE COURT: That's all your argument, but where's the claim?

MR. WEINGAERTNER: We don't have a copy of the claim in our slides today, but --

THE COURT: Well, why not? Somebody must have that. All right. Does somebody have the patent handy?

Oh, wait. I do have the patents. Okay. Which patent are we talking about here?

MR. WEINGAERTNER: Your Honor, it's the '104 reissue patent.

THE COURT: Is that it right there?

MR. WEINGAERTNER: Yes, your Honor.

THE COURT: '104.

(Reading)

"Apparatus comprising a memory containing intermediate form object code, constituted by a set of instructions, certain of said instructions containing one or more symbolic references and a processor configured to execute said instructions containing one or more symbolic references by determining a numerical reference corresponding to said symbolic reference,

storing said numerical references, and obtaining data in accordance with said numerical references."

Now, ordinarily, just without the benefit of anything, I would read that to mean -- okay -- symbolic reference is, like, *Y*; and numerical reference is, like, 7.

So do you have the instructions that would contain a reference to, say, *Y*? Or it could be *Y*1. And somewhere it's storing the number -- whatever number you want -- 99 -- to correspond to that symbolic reference. I don't know.

Why -- is that -- whose argument is that, or is that nobody's argument?

MR. WEINGAERTNER: I believe that's Oracle's argument, your Honor. And I think that --

THE COURT: What's wrong with that argument, then?

MR. WEINGAERTNER: What's wrong with the argument, your Honor, is that a juror could look at symbolic -- see numbers -- it could be actual numbers or numerals -- and think it's a numeric reference, but it's actually symbolic.

THE COURT: No. Wait. No. Wait, wait. But if you have the number 99 -- but see it says, further down there, "by determining a numerical reference corresponding to," so it's distinguishing between numerical and symbolic.

MR. WEINGAERTNER: I know that's right, your Honor; but if a juror were to see something that was actually -- that

contained numbers without any guidance, I think they could think that was numeric, but it wouldn't be. It would be merely symbolic. And I'd refer back to the baseball analogy.

THE COURT: But how could it be that a computer recognizes numbers as numbers? If you just got 99, it treats that as -- it doesn't treat it as a variable. It treats it as an absolute number; but if you have *Y*99, it treats that as a variable. You define it. It will recognize that. That's what I think. Anyway, it would recognize *Y*99 as a variable. You could then equate it to whatever you want it to be, for purposes of the next step in the process. And then maybe for the next step, you say, "Okay. Now it's going to be 150." And it varies. It's a variable. It's dynamic.

And -- but if you have just 99, is the way I understand computers work, it sees 99. It treats that as an absolute Arabic number. And -- you know, like from zero to infinity.

Isn't that the way it works?

MR. WEINGAERTNER: That's exactly the subtlety, your Honor, that we're trying to deal with here, because -- and again, that's why I used the baseball analogy. You could look at the back of somebody's uniform and see this number and think: Oh, that's numeric. And again, the metaphor is that: Where do they ultimately end up? Where does that data end up? It ends up somewhere in the computer statically.

THE COURT: But a baseball diamond is one thing, but the computer is set up to -- to recognize 99 as a real, absolute number. It's not treating that as a position on the baseball field.

MR. WEINGAERTNER: Well, I would disagree, your Honor. I could, if I wanted to, set up a variable: One, two, three. That just is a name of a procedure that I call "one, two, three." That does something else.

THE COURT: One hyphen, or one twenty-three?

MR. WEINGAERTNER: Oh, yeah. I could name the variable whatever I want to.

THE COURT: I don't think you can. I don't think -- unless it's in quotes or something, I don't think a computer will recognize one, two, three, as anything other than one hundred twenty-three.

MR. WEINGAERTNER: No. I think there's a distinction between the variable and the data that's ultimately pointing to. And that's -- it's a -- that's exactly, exactly what we think the problem is.

THE COURT: I don't think you can have a variable that's just straight out one, two, three, run together. Where does it say you can do that in this record?

MR. WEINGAERTNER: I don't think the patent necessarily says it. I think that what the patent does is provide for that situation by drawing a distinction between

dynamic and static.

THE COURT: But that language you showed me earlier -- go back to that language. Yeah, right there. It says, "static; i.e., numeric." So that's on one side of the equation -- not equation -- but one side of the distinction that's being drawn.

And then there's a big divide or -- versus dynamic and symbolic, are the ones that are on the other side.

So I -- to me, "numeric" means, like, 125 or 99.

Static -- it doesn't change, but dynamic is like a variable, like *Y*.

MR. WEINGAERTNER: Your Honor, I think -- I apologize.

THE COURT: It can't change. That's why it's called "symbolic." It's like *X* squared equals *Y* squared equals *Z* squared.

MR. WEINGAERTNER: The issue, though, in terms of static is where the data ultimately resides. And a variable that hasn't been resolved doesn't know where it's going to be yet. It's dynamic. It can move around. That's again why I used the baseball analogy. Everybody is ultimately a fielder on the field. And they have their position. And that position has a number. Pitcher is number one. Catcher is number two. That's fixed for all time.

Where an individual player might wind up, you don't

know. They have a number on their back for sure. That's a number on their back, but it doesn't tell them where on the field they're going to go.

So, similarly, a variable may have numbers in it, but you don't know exactly where it's going to be in the computer.

THE COURT: It's a pretty good jury argument. Again, I like that.

All right. We are running out of time. We've got about ten minutes to go. So do we have any more to construe here? I think there's one more, isn't there?

MR. JACOBS: On the '104 Patent, your Honor, it's "resolving."

THE COURT: Yes. Let's go to that one. We really got to -- you're -- you have about nine minutes left, Mr. Jacobs; but I don't see how I can get it done. I'm going to give you five minutes.

MR. JACOBS: Thank you, your Honor.

THE COURT: All right. Go ahead.

MR. JACOBS: Once again, we think the claim language is clear. And this -- in this case, "resolving" -- if you look at the claim language, it actually breaks out what is being -- what is meant by "resolving." And it varies from claim to claim as to how much elaboration and limitation on the "resolving" step is meant in each case.

And what Google wants to take -- do is take one of

those incidents of "resolving," and propose a construction based on it that has this notion of replacing.

We want -- again, we submit no construction is necessary, but at most, with -- with a construction of "resolving," "determining" is as far as one can go and get a fit to the claims and specifications.

THE COURT: You have that word in their comment sections in the opposing code? Is that why you want the word "no construction"? You got that one there, too?

MR. JACOBS: We do have that word there, your Honor.

THE COURT: That's what I figured. See how transparent all of this is? It's -- all right.

MR. JACOBS: And we'll show it to you. We're transparent. Well, almost transparent.

If we go to Claim 20, for example, Claim 20 says we're going to resolve the symbolic reference -- there's that expression again -- in the instruction by determining a numerical reference corresponding to the symbolic reference.

THE COURT: Let me just see. I want to read it out loud. Is this one of the ones at issue?

MR. JACOBS: Sure. It's one of the ones at issue.

THE COURT: Twenty?

"A computer-implemented method for executing a compiled program containing instructions in an intermediate form code,

at least" --

Let me just think about that. All right.

-- "at least one of the instructions containing a symbolic reference" --

So, like, that would be what I was saying a minute ago, like why, right?

MR. JACOBS: Yes.

THE COURT: (Reading)

-- "said method comprising the steps of resolving the symbolic reference in the instruction by determining a numerical reference corresponding to the symbolic reference."

Well, let me just stop for a second. I know that way before this patent ever came along, computer programmers were able to -- the computers would find if you were using *Y* or *X*1; whatever was your variable. It would go determine a numerical reference. That can't possibly have been a new invention.

So -- but what you're telling me is that you say that's what it means, is doing that old, traditional step in this -- in the context of this lead-in language that you have up there?

MR. JACOBS: Yes, your Honor, with one qualification. I think I'm going to help Google's counsel out here just slightly. When we're talking about numeric reference in

this -- in the context of these claims, we are talking about a numerical reference to a location in memory.

THE COURT: Oh. So you're not -- so it's not, like, one twenty-five.

You're saying it does have to be, like, a number in the -- a number in the stack, or whatever the right word is.

MR. JACOBS: Yeah. Slot 2, again, by analogy; by reference to the specification.

THE COURT: By analogy.

You're saying it has to be -- let me see if I get that. Okay.

-- "comprising the steps of resolving the symbolic reference in the instruction by determining a numerical reference" --

So this would be like Slot 2 or Slot 125?

MR. JACOBS: Exactly.

THE COURT: (Reading)

-- "corresponding to the symbolic reference."

So it could return something other than a number? In your view, then, it could return a number? It could return any kind of a symbol?

MR. JACOBS: In the numerical reference, it could return a pointer. It could return something that tells you where in memory to go; whereas *Y* is completely agnostic, if you will, as to where it sits in memory.

And the key is, if you want to -- the textual support for this in this claim language is: We're going to perform an operation with the instruction and data obtained in accordance with the numerical reference -- and then this is the heart of it in the context here -- without recompiling.

So we're going -- in this computer-implemented method for executing intermediate form code, we're going to resolve the symbolic reference to get the numerical reference location and data, and then we're going to -- then we're going to perform the operation in accordance with --

THE COURT: But you still agree that the red, highlighted language there -- that was in the prior art?

MR. JACOBS: Resolving symbolic references in instructions by determining numerical references?

THE COURT: Yes.

MR. JACOBS: On a stand-alone, basis? Yes.

THE COURT: All right. So then go ahead.

MR. JACOBS: So on this theme, though, that the concept of resolving is explicated in each of the claims in varying ways is evidenced by Claim 12, where we're going to resolve. And it's going to have two substeps. It's going to determine; and in this case, it's going to store.

Now, recall Google's proposed construction. Their proposed construction is "replace." They want to overwrite the original instruction. And they want to limit "resolving" to

this replacement.

There is a claim in which replacing occurs: Claim 14. It's a dependent claim. The method of Claim 3, wherein said substep of storing comprises the substep of replacing -- but if "resolving" always means "replacing," then we have a huge claim-differentiation problem. There's no purpose to Claim 14.

THE COURT: Well, what more does it mean, beyond replacing?

MR. JACOBS: Pardon me?

THE COURT: What else could it be, other than to replace?

MR. JACOBS: It could be to hold somewhere else in memory, but not overwrite.

As long as we rely on the "resolved" reference, we don't have to -- we don't have to erase the original variable. We don't have to erase the symbolic reference. As long as the computer knows I'm going to now go to the third house on the left, I didn't have to wipe out 36 Bulkley from my home address.

THE COURT: All right. Time's up.

MR. JACOBS: Okay. Thank you, your Honor.

THE COURT: Last word.

MR. WEINGAERTNER: Thank you, your Honor.

We hope to construe the configured medium, but we

maybe don't have a chance on this go-around.

So the points that were made by Oracle's counsel have to do with the presence of reading and rewriting together in the specification -- I'll just grab my pointer -- as showing some sort of distinction between them, which would be fine, except for the prior art that's admitted in the patent.

The prior art that's admitted in the patent basically has to do -- and this was actually something that came up during the tech tutorial. Counsel for Oracle basically was forced to admit that the hybrid solution -- this is this patent that we're talking about here -- is to -- quote, unquote -- "resolve" this symbolic reference; but the next time around, subsequent times rely on the previous resolution, which requires -- at minimum, requires storing of some sort; and we believe requires replacing.

The construction that's been presented doesn't have that in there at all. There's nothing to do with that.

It's basically determining. And who knows really, even, what that means. In fact, "determining" was actually not used in the original filing.

THE COURT: So wait. Go back to your -- you're saying, then --

MR. WEINGAERTNER: What we're saying, your Honor, is that the only way to fairly read this term -- and that the claim that was shown earlier that raised potential issues of

prior art -- that they really do raise those issues. And this is why Patent Office has found a substantial new question on precisely that language. The only way to really understand it is by looking at this figure and considering the prior art; that "resolving" in a vacuum, without any kind of replacement or rewriting at -- and this, by the way, is the core. This is really the entirety of the disclosure of what's allegedly patentable here. It's -- it's really hard to distinguish between -- between "resolving" and "rewriting." They seem to be the same thing.

Sure, they break them out into different words. That presents an issue. We recognize that issue, but we feel the only way to really deal with it and resolve, so to speak, is by recognizing this; by recognizing that the prior art doesn't allow you not to actually write it down and replace; and that, in fact, what's interesting is that there were claims in the patent -- Claims 12 and 13 -- that actually define "resolving" as including storing; but yet the definition that's proposed again miraculously disappears from the definition. It doesn't require it.

So we're not sure. We actually believe, your Honor, at some point, we will hopefully have a chance to explain to your Honor some of the real problems with the way the claims -- during this reissue process -- it stores the claims that go to all kinds of nuances that are not actually shown here. There's

something very interesting going on with these claims that I think leads back to this problem: What does it mean to resolve?

The only way we think it can be fairly construed is by taking into account that you have this rewriting; that you don't have this problem of having to do deja vu over and over again, and figure it out over and over again. You have to somehow replace it.

THE COURT: Okay. I've got a couple of questions for you all; both of you.

Do you still think that on the first one we discussed, there is at least a 50:50 chance you could agree on a meaning?

MR. WEINGAERTNER: I do, your Honor. And I think it comes down to including some notion of a file in that term; and that we could agree to that claim construction.

THE COURT: All right. Well, if I give you until -- today is Wednesday. If I gave you until Monday at noon, could you let me know by then, if you've been able to do that?

MR. WEINGAERTNER: Yes, your Honor.

MR. JACOBS: Yes, your Honor.

THE COURT: All right. And the other question is: What is the -- I know what the effect is. We can push right ahead. The reëxam, I mean. What do you expect is going to happen in the reëxam?

MR. WEINGAERTNER: Well, your Honor, just a day or two ago, the Patent Office in the '720 Patent issued an Office Action. An Office Action rejected all of the patents that are in play over all of the references that were submitted. And so at some point within the next two months, Oracle, if at all, needs to respond either with argument that would be directly affecting claim construction; at least, on that patent.

And the other patents are all in play also. We're expecting rejections on those, because the Patent Office found substantial questions on all of them. So they're going to be trickling in; we don't know exactly when. And so it raises a serious issue about whether or not your Honor is construing in the absence of all of the intrinsic evidence that will ultimately be there.

THE COURT: Well, the intrinsic evidence is already there.

This is -- but there could be further admissions made. And, of course, the PTO might narrow some of the claims -- I don't know -- or invalidate them.

Mr. Jacobs, what do you say?

MR. JACOBS: I think if memory serves me, having read your Honor's orders, I believe your claim constructions have a certain provisional quality to them; that -- that if -- that you rule based on the best information available to the Court as of the moment; but if something comes up later that should

cause that to be upset, the Court understands that that may have to be revisited. I think --

THE COURT: Yes, I do have that language, but I expressly say the lawyers can't ask to revisit it. Never, never -- but I can on my own, because I've learned, after doing ten patent trials, that almost every one of them, I sit there thinking: I wish I had known that before I made that ruling. I would have done a little bit better job on the claim construction.

So I keep that little right for wiggle room, but if I don't say it expressly, then the lawyers will say, "Well, Judge you wanted to reconsider, so here's a new argument you didn't consider before."

So, no, you can't do that.

MR. JACOBS: We're not unhappy with holding claim construction in abeyance while the reëxaminations proceed.

THE COURT: But then the case would languish.

MR. JACOBS: No. The case would proceed. And at some point, we're going to have to take a snapshot of where we are. And Mr. Weingaertner and his client would re-brief, based on the intrinsic record that has been created between now and then.

The alternative is -- I mean, I think there's a technical point on which his argument is correct, which is that there is additional administrative record being developed. We

don't agree that there should be a stay.

We don't necessarily urge the Court to put it in abeyance and not rule on the basis of the record the Court has before it, but I think we have to give room for allowance that either side might point to the intrinsic record over the next few months that's developed in the reëxam and say, "See? We were really right all along."

THE COURT: Well, our trial date is in time. I have one law clerk that I want to dedicate to this case. And, come November or end of November, she'll be gone. So we are going to have this case on its way to the Federal Circuit before that happens. So I'm not going to wait.

Most likely, I rarely would stay it based on reëxam, but I won't say "Never." I would have to -- it would have to be a very good case that I would save some time. Otherwise, I have to have a brand new law clerk learn this. You know, look at all of these lawyers you have out there. You have the luxury of resources that I don't have. And then if -- you cannot imagine what a burden a case like this is, to try to learn it well enough. And then when my law clerk leaves, then I have to -- I'm on my own, and have to come back up to speed. There's a huge burden.

So I -- you -- this is a very real factor that I'm being candid with you about that most judges wouldn't admit to, but it's a real factor. So if I was you, I would be thinking:

Whether that judge is crazy and gets it right or not, some jury's going to be sitting over there, come October, hearing you all point to the code that says "symbolic reference" -- or vice versa; whatever it is. Both of you have good points to make. And you'll be able to make them to a jury.

And, of course, by this point, you'll be down to one or two claims, because no good trial lawyer would go to a jury with seven patents and 123 claims. You go with two claims.

MR. JACOBS: Understood, your Honor.

THE COURT: So you'd better start thinking of how you're going to prioritize it.

So by Monday at noon, you'll please let me know if you've agreed. Otherwise, you're going to wind up getting what I've come up with.

MR. WEINGAERTNER: Yes, your Honor.

THE COURT: I want to say all of you are great. You're brilliant. You did a great job, and I'm very happy to have you here.

Okay. I do have a criminal case coming up in five minutes. I need to ask you to let the marshals come in, and the lawyers come in. Thank you.

MR. JACOBS: Thank you, your Honor.

(At 3:05 p.m. the proceedings were adjourned)

CERTIFICATE OF REPORTER

I, LYDIA ZINN, Official Reporter for the United States Court, Northern District of California, hereby certify that the foregoing proceedings in C. 10-3561 WHA, Oracle America v. Google, Inc., were reported by me, a certified shorthand reporter, and were thereafter transcribed under my direction into typewriting; that the foregoing is a full, complete and true record of said proceedings as bound by me at the time of filing.

The validity of the reporter's certification of said transcript may be void upon disassembly and/or removal from the court file.

_____

/s/ Lydia Zinn, CSR 9223, RPR

Monday, April 25, 2011