APPLE INC., a California corporation, Plaintiff, vs. SAMSUNG ELECTRONICS CO., LTD., a Korean business entity; SAMSUNG ELECTRONICS AMERICA, INC., a New York corporation; SAMSUNG TELECOMMUNICATIONS AMERICA, LLC, a Delaware limited liability company, Defendants.	CASE NO. 11-cv-01846-LHK

DECLARATION OF JOE TIPTON COLE IN SUPPORT OF SAMSUNG'S PROPOSED CLAIM CONSTRUCTION FOR U.S. PATENT NO. 7,698,711

I. ENGAGEMENT AND QUALIFICATIONS

A. Engagement

1. I, Joe Tipton Cole, have been engaged by Samsung Electronics Co., Ltd., Samsung Electronics America, Inc., Samsung Telecommunications America, LLC, (collectively "Samsung") in connection with the captioned lawsuit to provide my analyses and opinions regarding the interpretation of the term "applet" as used in the claims of U.S. Patent No. 7,698,711 ("the '711 patent").

2. The statements made herein are based on my own knowledge and opinion. I can and will testify to these matters if called at trial.

B. Qualifications

3. I have been actively engaged as a computer software professional since 1977. I am currently the owner and proprietor of Tipton Cole + Co., a professional practice of technical consulting services. The focus of the practice is litigation support for cases involving computer software and associated devices. From 1983-2008, I was President of Tipton Cole + Co., Inc., a software development and consulting company. That company prepared commercial database applications and provided both technical and management consulting services to large and small businesses. From 1977-1983, I was the co-owner of software development and consulting business, Cole & Van Sickle, which we founded in 1977.

4. I began studying computer science in 1968 as a student in National Science Foundation mathematics programs at Southern Methodist University and Texas A&M University. I continued those studies at Washington University in St. Louis and at the University of Texas at Austin, with a focus in the field of Artificial Intelligence.

5. I received a B.A. in Mathematics from the University of Texas in 1974, an M.A. in Computer Science from the University of Texas in 1976, and a Juris Doctorate from the University of Texas in 1978. I am currently licensed by the State Bar of Texas. My license status is “inactive.”

6. I have taught a Data Structure course at the University of Texas and a senior-level Database Management Systems Course at St. Edwards University in Austin.

7. I am being compensated for my time spent on the case at a rate of $675/hour.

8. My current CV is attached to this report as Exhibit 1.

C. Person of Ordinary Skill in the Art

9. I understand that to determine the ordinary and customary meaning of a claim term, one looks to the meaning that a person of ordinary skill in the art would have given the term at the time of the invention, which in this case is at least as early as August 30, 2005. Based on my experiences and the materials I have reviewed, it is my opinion that one of ordinary skill in the field of the '711 patent is a person in the field of computer science/engineering with either a Bachelor's degree and several years of relevant experience, or a Master's degree with less relevant experience, or a person with equivalent industry experience.

D. Background

10. Plaintiff and Defendant propose different constructions for the term *applet* used in claims 1, 9 and 17 of the '711 patent. The parties' constructions of the term *applet* are as follows:

Samsung's Proposed Construction	**Apple's Proposed Construction**
A small application designed to run within another program.	An operating system-independent computer program that runs within an application module.

11. The parties agree that an applet runs "within" another piece of software ("another program" or "an application module," respectively.) Given this

agreement, I focus my analysis on the question whether the term *applet* as used in the patent necessarily includes the limitation that the applet is “operating system-independent.”

E. Exhibits

12. Exhibit 1 is my CV.

13. Exhibit 2 is U.S. Patent No. 7,123,945, issued to Takeshi Kokubo, titled “Task Display Switching Method, Portable Apparatus and Portable Communications Apparatus."

14. Exhibit 3 contains relevant portions of the deposition transcript (rough) of Mr. Moon Sang Jeong, taken on November 17, 2011.

15. Exhibit 4 is a page from Mr. Jeong's notes with bates label SAMNDCA00139800.

16. Exhibit 5 contains relevant portions of *Wiley's Electrical and Electronics Engineering Dictionary* (2004).

17. Exhibit 6 is a copy of the complete web page from http://www.memidex.com/applet as of 2011 Nov 24, discussing the use of applet in various contexts.

18. Exhibit 7 is a compilation of documents that discuss the use of *applets* in the Microsoft Control Panel tools and Microsoft environment more generally. These documents include the following web pages: http://www.techimo.com/forum/applications-operating-systems/123490-windows-9x-question-about-32-bit-16-bit-software-installation.html (1998); http://www.codeproject.com/KB/winsdk/cjbcontrolpanelapplet.aspx (2000);

http://forums.windrivers.com/archive/index.php/t-16754.html (2001); http://www.pctools.com/guides/registry/detail/522/ (2002); http://www.realgeek.com/forums/control-panel-applet-missing-299659.html (2003); http://www.geekgirls.com/windowsxp_controlpanel.htm (2004); http://www.freewarefiles.com/Microsoft-Color-Control-Panel-Applet_program_16303.html (2005).

19. Exhibit 8 is a compilation of documents that discuss the use of *applets* in the context of AppleScript; a system of scripting language for the Macintosh OS X operating system. These documents include the following web pages: http://managingosx.wordpress.com/2006/03/23/universal-applescript-applets/ (2006); http://fm.geckotribe.com/applescript/savenonpopup.php3 (2001); http://www.applefritter.com/node/15241 (2006); http://macscripter.net/viewtopic.php?id=24382 (2002); and http://www.mactipsandtricks.com/articles/BAS1.lasso (2004).

20. Exhibit 9 is a compilation of documents that discuss the use of *applets* in the Linux environment. These documents include the following web pages: http://linux.softpedia.com/get/Desktop-Environment/Tools/GetCodecs-5182.shtml (2005);http://www.pygtk.org/articles/applets_arturogf/ (2004); http://ubuntuforums.org/showthread.php?t=909404.

21. Exhibit 10 is a webpage that discuss the use of *applets* in the Ruby programming language for Windows-specific environments. The webpage can be found at: http://www.justskins.com/forums/anyone-using-registerclass-from-33437.html.

22. Exhibit 11 is a webpage that discuss the use of *applets* in the Flash programming language. The webpage can be found at: http://scripts.top4download.com/stock-applets-script/efiow.html.

23. Exhibit 12 is a webpage discussing the ability of programmers to overcome the default restrictions in Java, thereby gaining access to the operating system of that particular machine. The webpage can be found at: http://www-personal.umich.edu/~lsiden/tutorials/signed-applet/signed-applet.html. (The copyright date on the document at the "jarsigner" link is 1994-2004.)

II. THE MEANING OF "APPLET"

A. Intrinsic Evidence

24. There is limited discussion in both the specification and claims of the '711 patent, as well as the prosecution history, as to the definition of the term "applet."

25. While the discussion of "applet" is limited, Samsung's proposed claim construction is fully supported by the intrinsic evidence. The intrinsic evidence makes clear that the term "applet" is not limited to an "operating system-independent computer program that runs within an application module" as proposed by Apple and its expert, Dr. Givargis. Apple's reliance on the use of the term "applet" in the Java language is wholly unsupported by the '711 patent claim language and specification, as well as the entirety of the '711 patent's prosecution history.

1. The '711 Patent Disclosure

26. The '711 patent describes an "apparatus and method capable of performing multiple tasks in a portable terminal…in which the menu functions of

the portable terminal can be implemented while continuing to play the music." ('711 Patent Abstract).

27. The apparatus and method of the '711 patent were invented to deal with the problem that portable terminals had with allowing users to multi-task while listening to music in the background. As embodied in the prior art, users could not "simultaneously work on several menus of the portable terminal while listening to the music using the conventional MP3 music function." ('711 Patent at Col. 1:39-41). The '711 patent solved this problem by disclosing an apparatus and method by which "menu functions of the portable terminal can be implemented while continuing to play a music file." ('711 Patent at Col. 1:58-61).

28. When discussing the types of multi-tasking the '711 invention could perform while listening to an MP3 file, the specification lists, as examples, many different types of applications, including messaging, phone book, scheduling, games, and picture searching. ('711 Patent at Col. 4:43-49). Additionally, dependent claims 7-8 and 15-16 specifically mention utilizing messaging and phone-book functions while continuing to play MP3 files in the background. ('711 Patent at Claims 7-8 and 15-16).

29. The wide variety of applications that can be implemented while playing an MP3 file contradicts Apple's narrow construction of *applet* as only referring to a Java applet being solely utilized in a web browser environment. There is no evidence that any of the disclosed functionalities in the '711 patent use the Java programming language or have any relation to a web browser.

2. Specification and Claims

30. The term “applet” appears in the three independent claims of the ‘711 patent, claims 1, 9 and 17. In all three claims, “applet” appears in the same context, namely:

> “a music background play object, wherein the music background play object includes an application module including at least one applet.”

31. As used in claims 1, 9 and 17, the term “applet” is never once qualified with the notion that the “applet” is Java based, nor that the “applet” is operating system independent.

32. In fact, the only instance where the term “applet” is discussed in the specification also fails to qualify the term with any requirement that the “applet” be written in Java or be operating system independent. This portion of the specification states:

> “FIG.1 is a block diagram of a portable terminal according to an exemplary embodiment of the present invention, in which an MP3 music control processor is not included. Application modules of the portable terminal include at least one applet and each of the application modules, that is each menu of the portable terminal, independently performs multi-tasking.” (‘711 patent at Col. 3:8-14)

Neither the language quoted here, nor FIG. 1 of the ‘711 patent, even mention much less require the “applet” to be operating system independent as Apple and Dr. Givargis propose.

3. File History

33. The claim language that includes the term “applet” was added at the request of the patent examiner. As detailed in an interview summary, the “[e]xaminer suggested to further include the definition ‘a music background play object’ as ‘wherein the music background play objects including an application module includes at least one applet’ as argued during the interview to distinct [sic] from the icon as taught by KOKUBO.” (U.S. Patent Application No. 11/778,466, Examiner’s Interview Summary of December 16, 2009).

34. As a result of this request, the claims were amended to include the language suggest by the patent examiner. The language added to claims 1, 9 and 17 was “wherein the music background play object includes an application module including at least one applet.”(U.S. Patent Application No. 11/778,466, Applicant’s December 8, 2009 Arguments/Remarks Made in an Amendment at pp. 2-4).

35. While Dr. Givargis is correct when it states that there was no definition of *applet* proposed either by the examiner or the Applicant, Dr. Givargis fails to analyze the one piece of prior art that this amendment was aimed at distinguishing. This piece of prior art is U.S. Patent No. 7,123,945, titled “Task Display Switching Method, Portable Apparatus and Portable Communications Apparatus (the “Kokubo patent” attached hereto as Exhibit 2). Generally speaking, the Kokubo patent was aimed at allowing the “processing of a plurality of tasks in parallel and of displaying a plurality of display regions for displaying data.” (Kokubo patent at Abstract). The Kokubo patent does not deal with the use of Java or applets in order to achieve multi-tasking.

36. The Kokubo patent discloses an apparatus and method that solved the problem of multi-tasking by using icons to switch between applications faster than simply closing out of an application to perform another function. (*See* Figs. 3-9, Col. 2:29-62). The icons would also correspond to a "predetermined state" of the corresponding application, such as a "suspended or stopped state," which would further increase the speed of switching between operations. (*Id*. At Col. 3:33-41).

37. The Kokubo patent did not disclose, or even discuss, operating system independence, or dependence, of any of the applications or programs run with the disclosed multi-tasking method. Given that Kokubo is silent on the use of operating system dependent or operating system independent, it is not reasonable to argue that the addition of *applet* to the claims of the '711 patent somehow requires an "applet" as used in the claims of the '711 patent to be an operating system independent piece of software.

4. Conclusion

38. Given the limited amount of intrinsic evidence to construe the term *applet*, I have examined extrinsic evidence to determine the general meaning of *applet* at the time of the invention. In my opinion, the proper construction of the term *applet* in the context of the '711 is "a small application designed to run within another program." "Applet," as used in the '711 patent, should not include the limitation "operating system-independent." My opinion is based on the extrinsic evidence that I present below and is consistent with the intrinsic evidence. This evidence consists of various uses of the term "applet" during the time period

leading up to the ‘711 patent’s priority date of August 30, 2005, as well as the testimony and contemporaneous documentation of the sole inventor of the ‘711 patent, Mr. Moon-Sang Jeong. This evidence shows that to one of ordinary skill in the art the term *applet* was not limited to “operating system-independent” programs.

B. Extrinsic Evidence

39. Before discussing the evidence that one of ordinary skill in the art would not consider all uses of the term “applet” to be operating system independent, the testimony of Mr. Jeong and contemporaneous documents authored by Mr. Jeong provides context to the term “applet” as used in the ‘711 patent. At his deposition, Mr. Jeong was asked to give his definition of “applet” as he understood it in 2005. Mr. Jeong responded that his “understanding as to an applet was in reference to such things as would comprise an application, such as: smaller functionalities, smaller classes of things, even a smaller unit of an application or of applications.” (11/17/2011 Deposition of Moon-Sang Jeong, Rough Tr. at 25:2-13, attached hereto as Exhibit 3).

40. Mr. Jeong testified that the term “applet” can be used in both an operating system independent fashion, as well as an operating system dependent fashion. (Exhibit 3 at 25:14-25). Although “applet” could be used in both ways, Mr. Jeong testified that he was using “applet” in the operating dependent fashion due to his development of the Qualcomm platform that only use Qualcomm chipsets. (Exhibit 3 at 29:10-30:22). When “applet” was used in this context, the term

"applet" "was something that applied with respect to the Qualcomm platform only." *Id*.

41. Mr. Jeong's testimony is also consistent with the documentation that he created during the development process of the invention that would eventually issue as the '711 patent. Mr. Jeong's development notes contain an entire page dedicated to identifying applets and interfaces, none of which are identified as being Java applets or operating system independent. (*See* SAMNDCA00139800, attached hereto as Exhibit 4). The identified applets include functions such as "Clockdesign," "ScheduleApp," "Mediaplayer," "QCamcorder," and many others that neither reference Java nor being used in an operating system independent environment.

42. Mr. Jeong's contemporaneous documentation supports his testimony, as does the large quantity of extrinsic evidence which shows that "applet" does not require operating system independence.

43. *Wiley's Electrical and Electronics Engineering Dictionary* (2004) defined "applet" as a "[a] small application designed to run within another program." This definition comports with Samsung's definition of *applet* and makes no mention of the requirement that *applet* be an operating system independent computer program.

44. In addition to this dictionary definition, Apple and Dr. Givargis cite another dictionary that supports Samsung's construction of the term *applet*. Dr. Givargis cites the *McGraw-Hill Dictionary of Scientific and Technical Terms* (6th Ed., 2003) at page 124, which defines *applet* as "a small program, typically written in

Java*.”* (Givargis Decl. at ¶ 42). Even though this definition refers to Java, it does not require that the *applet* actually be written in Java, or that the applet be operating system independent. Given Apple's and Dr. Givargis' absolute position that an *applet* must be operating system independent, Apple's and Dr. Givargis' cited definition directly contradicts such an absolute position, and supports Samsung's construction that *applet* does not require operating system independence.

45. Exhibit 6 illustrates the challenge of presenting a precise, authoritative definition of the term *applet*. Various of the dictionary sources shown in the "Definition references" section of that page limit the term *applet* with the qualification: 1) small; 2) simple; 3) automatically copied; 4) run from a webpage; and 5) **usually** portable between operating systems. Four of the nine include a notion of *inclusion* similar to the "application module including at least one applet" phrase that appears in the claims, the specification and the file history of the '711 patent. As a point of interest, however, none of the definitions limits the term *applet* to the realm of *Java applets*.

46. The "Etymological references" section of Exhibit 6 offers to definitive dates, 1990 and 1995, as the date of first use. The 1990 date given by Merriam-Webster matches the date attributed to the Oxford English Dictionary at the Wikipedia page http://en.wikipedia.org/wiki/Applet. The date has some significance because Java was unknown in 1990, so that the term *applet* would have a meaning independent of the phrase *Java applet*.

1. As Used in the '711 Patent, *Applet* Does Not Refer to a Java Applet

47. Dr. Givargis focuses on definitions of *applet* that are somehow related to the Java programming language. The only evidence Dr. Givargis presenst to support this logical leap is that "[l]eading up to 2005, mobile phone manufacturers increasingly produced Java-enabled devices." (Givargis Decl. at ¶ 22). Under Dr. Givargis' analysis, the mere fact that some mobile phones were Java-enabled at the time of the invention would require that any mention of *applet* in a patent for mobile devices must refer to a Java applet. The logical fallacy of this conclusion is obvious.

48. Neither the patent nor the file history invoke a Java-based environment as a necessary element of the '711 patent. In fact, the word "java" never occurs in the patent itself.

49. Those of skill in the art would not make this logical leap. For those of skill in the art, there is a more expansive view of the term *applet*, one that encompasses more than just *Java applets*.

50. In addition to Java applets, *applet* was used to refer many different programming languages and functionalities. Some of these non-Java related references to an *applet* include:

a. Desktop Applets

51. Use of the term *applet* to describe what Microsoft sometimes calls "Control Panel Tools" was current for some time before and after the filing of the '711 patent. (*See* Exhibit 7)

b. AppleScript Applet

52. AppleScript is a system scripting language used for the Macintosh OS X operating system. AppleScript uses *applets* in its programming language. (*See* Exhibit 8)

c. Linux Applets

53. The Linux operating system also used the term *applet* extensively in its environment.

54. *Applet* is a term that is also used in the Python programming language for the Linux operating system. An example of one such applet is the GetCodecs applet (*See* Exhibit 9) Another example of Python *applets* is a collection of Gnome (Linux-specific) applets written in Python. (*Id*.)

55. Not only did Linux use the term *applet,* there were uses of *applet* in reference to specific versions of Linux, such as Ubuntu. (*Id.*)

d. Ruby Applet

56. In addition to the earlier referenced Microsoft desktop applets, another example of a Windows-specific *applet* is an *applet* written in Ruby for the Windows operating system. (*See* Exhibit 10)

e. Additional Applets

57. Additional uses of the term *applet* can be found in the Visual Basic programming language as well as Flash. (*See* Exhibit 11).

58. It is my opinion that given the wide ranging use of the term *applet* in the field of the '711 invention, one of skill in the art would not limit the term *applet* to refer solely to a Java applet. Add to this simple fact, the silence of the '711

patent and file history as to any use of Java, it is not correct to equate the term *applet* to the more restrictive *Java applet*.

2. *Applet* Does Not Have to be Operating System Independent

59. As discussed in the previous section, the term “applet,” unadorned with any qualifiers, is subject to many “definitions.” Taken in isolation the vast majority of the definitions do not address the issue of operating system dependence or independence.

60. Dr. Givargis cites no evidence from the intrinsic record that would support limiting the term *applet* to require operating system independence. Dr. Givargis quotes extensively from the file history of the ‘711 patent and asserts that the many passages support his opinion, but he never explains why that is so. In my opinion, taken individually, as a whole, or in any combination, the passages on which Dr. Givargis relies offer no support for his conclusion. In the context of the ‘711 patent, because these references are utterly silent on the question of operating system independence, they contradict Dr. Givargis’ stated opinion.

61. Apple’s own extrinsic evidence also contradicts the conclusion that as used in the ‘711 patent, *applet* means an “operating system independent computer program.” Dr. Givargis places a large emphasis on the *Java’s Developer’s Resource* (1997) by Eliotte Harold (“the Harold reference”). While the selection of a reference that deals specifically with Java is inherently flawed when there is no indication that *applet* as used in the ‘711 patent refers to a Java specific applet, even this Java specific handbook recognizes that an *applet* can be operating system dependent. According to Apple, Harold explains “how

applets **can** be independent of the host platform." (Givargis Decl. at ¶ 45). As stated, this conclusion falls short of Apple's proposed construction in which applets **must** be independent of the host platform.

62. Furthermore, the Harold excerpt does not say that Java applets running within "Web browsers" are operating system independent. Neither does this excerpt support the conclusions that the web browsers are operating system independent or that web browsers, even if operating system independent, impose that condition on the applets that execute within them. Apple and Dr. Givargis fail to take any possibilities into account.

63. While Java applets may in practice be operating system independent, it takes particular skill and commitment to do so. The Java sandbox in which Java applets run protects the software author from many actions that deal directly with the host operating system – but operating system independence is not automatic. For example, the programmer must refrain from taking advantage of actions or data that are available only on one operating system. Some actions or data may be available on multiple operating systems but under different names or with different details of definition such as signatures of system procedures or data types of system variable or system constants.

64. Additionally not all Java applets are necessarily prohibited from exploiting the peculiar capabilities of their host operating system. A brief exposition of a mechanism to overcome the default sandbox restrictions imposed on Java was readily available to one or ordinary skill in the art. (*See* Exhibit 12).

65. As discussed previously, the term *applet* is used in conjunction with many different programming languages, and some of those applets are operating system dependent. There are specific *applets* for the Windows operating system, as well as *applets* for different versions of the Linux operating system. As these applets are used specifically with one operating system, they are the essence of operating dependent *applets*. Apple should be well aware of the operating system dependent nature of the term *applet* as it uses *applets* specific to its own operating system.

66. In conclusion, Apple's requirement that an *applet* be operating system independent contradicts the plain language of the '711 patent and the file history, as well as Apple's own extrinsic evidence. At best it can be said that Java *applets* can be operating system independent, but there are instances where that is not the case. There are also many other *applets* that are used for specific operating systems and therefore by definition are not operating system independent. One skilled in the art would not so limit the term *applet* as to **require** operating system independence.

C. Opinion

67. In the Java-free context presented by the '711 patent, one of ordinary skill in the art at the time of the invention would not include the limitation "operating system-independent" as part of the definition of the term *applet*.

68. Based on the above intrinsic and extrinsic evidence, it is my opinion that one of ordinary skill in the art at the time of the invention would have interpreted the term *applet* as "a small application designed to run within another program."

I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct.

Executed on November 28, 2011, at Redwood Shores, California.

J. Tipton Cole