IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

TV GUIDE ONLINE, INC. and TV GUIDE ONLINE, LLC Plaintiffs, Counterclaim-Defendants, v. TRIBUNE MEDIA SERVICES, INC. Defendant, Counterclaim-Plaintiff	C.A. No. 05-cv-725-***-LPS

REDACTED PUBLIC VERSION

DECLARATION OF STUART W. YOTHERS IN SUPPORT OF TV GUIDE ONLINE'S REPLY BRIEF IN SUPPORT OF ITS MOTION FOR SUMMARY JUDGMENT THAT TMS INFRINGES CLAIMS 1-8 AND 10 OF THE '078 PATENT

OF COUNSEL:

Christopher J. Harnett
Ching-Lee Fukuda
Stuart W. Yothers
Ropes & Gray LLP
1211 Avenue of the Americas
New York, New York 10036
(212) 596-9000

Ronald E. Naves, Jr.
Jeffrey A. Fehervari
Gemstar-TV Guide International, Inc.
6922 Hollywood Blvd.
Hollywood, California 90028
(323) 817-4600

Frederick L. Cottrell, III (#2555)
Jeffrey L. Moyer (#3309)
Steven J. Fineman (#4025)
Richards, Layton & Finger PA
One Rodney Square
920 North King Street
Wilmington, Delaware 19801
Telephone: (302) 651-7700
cottrell@rlf.com
moyer@rlf.com
fineman@rlf.com

Attorneys for Plaintiffs TV Guide Online, Inc. and TV Guide Online, LLC

December 21, 2007

I, Stuart W. Yothers, do hereby declare:

1. I am a member of the bars of the State of New York and the Commonwealth of Massachusetts and am an associate of the firm of Ropes & Gray LLP, counsel for plaintiffs and counterclaim defendants TV Guide Online, Inc. and TV Guide Online, LLC (collectively, "TV Guide Online"), in this action. I submit this declaration in support of TV Guide Online's Reply Memorandum in Support of its Motion for Summary Judgment that TMS Infringes Claims 1-8 and 10 of the '078 Patent.

2. Attached hereto as **Exhibit A** is a true and correct copy of pages 14-21, 50-53, 102-105, and 214-217 of the transcript of Dr. Gary S. Tjaden's June 15, 2007 deposition.

3. Attached hereto as **Exhibit B** is a true and correct copy of pages 35-36 of the Rebuttal Expert Report of Dr. Gary S. Tjaden Regarding Non-Infringement of United States Patent No. 5,988,078, dated February 9, 2007.

4. Attached hereto as **Exhibit C** is a true and correct copy of the Declaration of Dr. Gary S. Tjaden, dated November 20, 2007.

5. Attached hereto as **Exhibit D** is a true and correct copy of the Declaration of Gregory Brian Loose, dated November 20, 2007.

6. Attached hereto as **Exhibit E** is a true and correct copy of TMS0002462-64 from TMS's document production in this case.

7. Attached hereto as **Exhibit F** is a true and correct copy of pages 246-249 of the transcript of Barbara Needleman's September 29, 2006 deposition.

8. Attached hereto as **Exhibit G** is a true and correct copy of pages 18-25 of the transcript of Kevin Walter McFall's August 9, 2006 deposition.

9. Attached hereto as **Exhibit H** is a true and correct copy of pages 14-17 of the transcript of Akira Mark Yamada's August 29, 2006 deposition.

10. Attached hereto as **Exhibit I** is a true and correct copy of pages 42-45 of the transcript of Michael Silver's September 8, 2006 deposition.

11. Attached hereto as **Exhibit J** is a true and correct copy of pages 14-17 of the transcript of Gregory Brian Loose's September 20, 2006.

12. Attached hereto as **Exhibit K** is a true and correct copy of pages 282-285 of the transcript of James D. Fehnel's October 25, 2006 deposition.

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

Executed in New York, New York,
on December 20, 2007

Stuart W. Yothers

EXHIBITS A & B

REDACTED IN THEIR ENTIRETY

EXHIBIT C

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

TV GUIDE ONLINE, INC. and TV GUIDE ONLINE, LLC, Plaintiffs and Counterclaim Defendants, v. TRIBUNE MEDIA SERVICES, INC., Defendant and Counterclaim Plaintiff.	Case No.: 05-725-***

DECLARATION OF GARY S. TJADEN

I, Gary S. Tjaden, declare and state as follows:

1. I have been retained by Tribune Media Services, Inc. ("TMS") as an expert witness in this action to provide my technical assessment and opinions regarding U.S. Patent No. 5,988,078 ("the '078 patent").

2. I am currently Founder and President of COCOMO ID, LLC, a developer of technology for mobilized speech-audio publishing.

3. I hold a Bachelor of Science degree in Electrical Engineering (B.S.E.E.), which I received from the University of Utah in 1966. I received a Master of Science degree in Electrical Engineering (M.S.E.E.) in 1969 from Northwestern University. In 1973, I received a Doctor of Philosophy (Ph.D.) degree in Computer Science from the Johns Hopkins University.

4. I have personal knowledge of all facts set forth in this Declaration, and could testify to these facts if called to do so.

5. The claim term **"controller being programmed to perform,"** is found in Claim 8 of the '078 Patent. TMS's proposed construction of the term is "computer application

programs are installed on the controller prior to its use by a user to perform." This declaration is concerned with TV Guide's objection to TMS's proposed construction of this term.

6. As I have stated in my Non-Infringement Rebuttal Expert Report, and in my October 5, 2007 Declaration in this matter, it is clear from the '078 specification that the functions performed by the "controller" are embodied in a computer application that runs on the controller (or personal computer-like device). At the priority date of the '078 patent (March 1992), personal computer application programs were completely installed on the personal computer before they began executing. The concept of downloading during the execution of the application program incremental functionality for it to perform was not used or commonly known to skilled artisans. Only several years after the introduction of the first widely available Internet browser, a time subsequent to the '078 patent's priority date, was this concept commonly known and in use. Thus, I conclude that all of the functions embodied in the '078 controller's programming must exist as one or more software executable files (computer application programs) within the controller prior to its use by a user to perform those functions. By this I mean that all functions embodied in the controller's programming must exist therein prior to its use by a user to perform *any* of them.

7. Mr. Cole, TV Guide's Expert Witness, responded to my above statements in his Supplemental Expert Report Of J. Tipton Cole Concerning Infringement and Validity of February 28, 2007 ("Cole Supplemental"), stating: "At the time of the '078 patent's priority date, examples of systems that downloaded or installed incremental functionality while the application program was being executed were common." (Cole Supplemental at ¶37.) Mr. Cole then listed three documents as examples of his assertion, attached to his report as Exhibits D, E and F, which I shall refer to here as "Goldszmidt," "Korb" and "Bell," respectively. Mr. Cole

did not cite to any specific passages in these documents, nor did he discuss their contents in any further way. When questioned about this in his deposition Mr. Cole did point to three passages in the Goldszmidt document and one in the Bell document that he claimed support his contention, but he did not elaborate as to how they support it. (Cole Deposition Transcript dated June 21, 2007 at pp. 244-247.)

8. In fact, the Goldszmidt, Korb and Bell documents referenced by Mr. Cole do not show examples of systems "that downloaded or installed incremental functionality while the application program was being executed," as he contends.

9. Before discussing the documents a bit of background regarding how personal computers function may be in order to assist in understanding the technology described in the documents. When one buys a new personal computer today, and when one bought a new personal computer in 1992, included with or within the computer are several hardware components and some pre-installed software. The hardware components include a central processing unit (CPU) having volatile memory (called main memory or RAM (random access memory)), a non-volatile memory (called a hard drive), a removable memory device (a floppy disk drive in 1992, or a CD (compact disk) drive today), a keyboard, a monitor, some input/output (I/O) ports, and, often, some input/output devices such as a telephone modem.

10. The software will include a Basic Input/Output System (BIOS, and usually called firmware because it cannot easily be changed) retained on yet another memory device called read-only memory (ROM), and an operating system (e.g., Microsoft Windows or Apple OS X). The operating system is stored on the computer's hard drive. Some application software, such as a wordprocessor, may also have been "installed" and is stored on the hard drive. When the user turns-on the computer the BIOS firmware program is activated. It is responsible for activating

and controlling basic system functions such as communications between the CPU and the other hardware devices. The BIOS also causes certain portions of the operating system software to be copied from the hard drive to the main memory. When this is completed control is transferred from the BIOS to the operating system and the computer is ready for use by the user.

11. Computer software is developed by software engineers using "programming languages" that allow the units of the software (statements or instructions) to be expressed in textual form. This is because the form in which the computer instructions that make up the program that the CPU actually executes is that of combinations of one's and zero's (bits or binary digits), something very hard for humans to comprehend. While comprehending a computer programming language requires much training and skill, doing so is much easier than comprehending instructions in the form of binary digits.

12. Once a computer program has been written in a programming language, it normally (and commonly in 1992) must be translated into binary digit form in order for a computer's CPU to execute the program (i.e., perform the functions specified by the program). The process of translating from a programming language to binary digit form is called "compilation." That is, a computer program written by a human in a programming language normally must be "compiled" by another computer program, called a compiler, before it can actually be executed by a computer. It is the binary digit form of the computer program that comprises the software retained on a computer's hard drive for subsequent copying to main memory when the program is to be executed.

13. Furthermore, in order for a computer program to be executed by a personal computer in binary digit form the program must first be "installed" on the personal computer's hard drive. This installation process involves both copying all of the necessary files in binary

digit form (called executable files) to the computer's hard drive, and "registering" certain information regarding the location of the files with the computer's operating system.

14. It is possible for a computer program in program language form to be executed by a computer without first being compiled. The process for doing so is called "interpretation." The program so executed is often called a "script." In order for such interpretation to occur a special program, called an "interpreter," must be installed (in binary digit form) on the computer. The interpreter, when presented with the script, performs the function of, in effect, compiling the script immediately before it is executed under control of the interpreter.

15. Use of scripts and interpreters, in my opinion, and as explained in detail in my Non-Infringement Rebuttal Expert Report, is outside of the scope of the '078 patent. As stated therein and repeated above, such use was not commonly known to a skilled artisan in 1992 and was not used in personal computers in user's homes as required by the '078 patent. Rather, in order for a computer program to have been executed on a personal computer in a user's home in 1992 the program would first have had to have been installed in binary digit form on the computer's hard drive.

16. At his deposition, Mr. Cole identified three passages in Exhibit D to his Supplemental Report, the Goldszmidt article, to support his opinion: "It supports the ability to extend the functionality of servers (agents) at execution time…" (Abstract); "Delegation provides a flexible way for managers to dynamically associate functionality and responsibility with other managers." (p. 350); and "From a software configuration perspective, delegation provides a simple yet powerful scheme to dynamically compose management systems, by connecting and integrating independent delegated scripts. It is not required, though allowed, that manager process actually write programs to be delegated during their execution." (p. 351) (Cole

Deposition Transcript dated June 21, 2007 at pp. 245-247) In my review of Goldszmidt, neither the passages identified by Mr. Cole nor any other portion of the article supports Mr. Cole's opinion that "At the time of the '078 patent's priority date, examples of systems that downloaded or installed incremental functionality while the application program was being executed were common." (Cole Supplemental at ¶37.)

17. The Goldszmidt document is entitled "Network Management by Delegation – The MAD Approach." It has three authors, one a Ph.D. candidate in the Department of Computer Science at Columbia University, another an associate professor at the Columbia University Computer Science Department, and the third a member of the research staff at the IBM Thomas Watson Research Center. The document is a research paper presented at the *1991 Conference of the Centre For Advanced Studies on Collaborative Research.* The paper deals with issues of the management (i.e., operation and control) of large corporate private communication networks. The first sentence of the Introduction section sets the stage, saying: "Industrial enterprises are increasingly dependent upon large scale complex networked systems serving as their information backbones." (Goldszmidt at p. 347, Introduction.) In the Abstract of the paper the authors say regarding its content and purpose: "We describe the delegation model, its application to network management, and the design of a prototype implementation." (*Id.* at p. 347, Abstract, emphasis mine.)

18. Two points can be made so far regarding Mr. Cole's interpretation of the relevance of the Goldszmidt paper to the issue at hand. First, since the paper is a research paper, and was published in late 1991, its contents can hardly be characterized as representing methods that would have been commonly known to a skilled artisan in 1992. Secondly, since the subject is dealing with the operation and control of corporate networks, whatever it might have to say

can hardly be understood as applying to the use of personal computers in user's homes as required by the '078 patent and as so understood by a skilled artisan.

19. Goldszmidt sets the context for its teaching by saying: "A typical network management system [Rose91] involves agents, which are responsible for monitoring and controlling managed-objects of the network, and managers, which have the responsibility for collecting dynamic status data from the agents, interpreting that data, and directing the agents (e.g., how to handle fault scenarios)." (*Id.* at p. 347, col. 2, para. 2.)

20. Goldszmidt describes the problem addressed in the paper as follows: "Agents and managers must spend bandwidth and cycles to divide detection responsibilities. ... However, both SNMP and CMIP provide limited or no capabilities to combine primitives to handle composite scripts. Thus, the manager must step the agent through the execution of the script." (*Id.* at p. 349, col. 2, para. 5 – p. 350, col. 1, para. 2.) Goldszmidt refers to this as the "micro-management problem." ("SNMP" and "CMIP" are acronyms for protocols used by managers to communicate with agents.) And, Goldszmidt proposes that: "The micro-management problem can be resolved by delegating entire composite management scripts from managers to agents. ... The agent would then execute the composite script without manager interaction (except when the script itself requires it)." (*Id.* at p. 350, col. 1, para. 6 – col. 2, para. 1.) Goldszmidt refers to its proposed solution throughout the paper with the acronym "MAD," which stands for "Manager-Agent Delegation."

21. The above "composite script" is referred to in Goldszmidt as a "Delegated Management Program" (DMP). "A DMP is written in a Management Scripting Language (MSL). There is a whole spectrum of languages that can serve as MSLs, from general purpose programming languages such as C, to more restricted specialized languages for management."

(*Id.* at p. 351, col. 2, para. 3.) To use a DMP it is "... transferred via a delegation protocol, from a delegating manager to an accepting agent, which stores it in the Repository." (*Id.* at p. 351, col. 2, para. 1.) "The Repository allows storing and retrieving of DMPs from which the DMPIs [*Delegated Management Process Instances*, or executing programs] can be instantiated. DMPs can be stored at agent startup (boot time), or downloaded from a remote manager into the Repository during execution, via the delegation protocol. After receiving a Delegate request, the agent will store the DMP in a repository, and return to the delegating manager (through the protocol) a handle to it (DMPIid)." (*Id.* at p. 353, col. 2, para. 9.)

22. In other words, the above paragraph says the following: A computer program (DMP) is written in a programming language (MSL). The program is for execution on an agent. The program is copied to the hard drive of the agent (Repository). It can be copied to the hard drive when the agent is not running (e.g., during factory setup or boot time), or while the agent is running another program (during execution of), say, a wordprocessor. After the program is copied to the hard drive it is "installed" on the agent (producing the DMPIid necessary for execution, or "instantiation," to be initiated).

23. Regarding the process I have called "installation," Goldszmidt provides the following details for its MAD system: "Each DMP is represented in the repository as a data structure which contains the actual code of the program, and additional information describing its characteristics and requirements. ... DMPs can be in source code format, which may require compilation and linking or interpretation, or as object code that requires only dynamic linking. The Mad Agent uses the Translator for preparing the DMP for instantiation, via compilation and linking. The Translator uses the Repository to store the executable images which are produced by the compilation." (*Id.* at p. 354, col. 1, para. 1.) Here Goldszmidt proposes a type of

installation that was certainly not used on personal computers in user's home in 1992, and still is not used today, that is, compiling a program locally on the personal computer during the installation process. Goldszmidt does refer to interpretation in passing, but again only if the program to be interpreted is already "installed" in the Repository (agent hard drive). Such interpretation was not used in the prototype system described by Goldszmidt, nor is it performed by user's personal computers when using the Zap2it.com service.

24. The above discussion of Goldszmidt should make it clear that the Goldszmidt document actually affirms my contention that at the priority date of the '078 patent (March 1992), personal computer application programs were completely installed on the personal computer before they began executing. In any case, Goldszmidt's teachings would apply only to systems of communicating computers used to operate and control private computer networks, not to consumer applications used in user's homes.

25. I turn now to the Korb document, Exhibit E to Cole's Supplemental Report. At his deposition, Mr. Cole was unable to identify specific passages that support his opinion. (Cole Deposition Transcript dated June 21, 2007 at pp. 247-248). In my review of Korb I do not find anything to support Mr. Cole's opinion that "At the time of the '078 patent's priority date, examples of systems that downloaded or installed incremental functionality while the application program was being executed were common." (Cole Supplemental at ¶37.)

26. Korb is a research paper written by two authors who, at the time, were associated with the Department of Computer Science at Purdue University. The paper does not give their positions, but typically one would be a student and the other the student's research advisor. The paper, entitled "Command Execution in a Heterogeneous Environment," was published in 1986 in the Proceedings of the Association For Computer Machinery (ACM) Special Interest Group

on Communications (SIGCOMM) *Conference on Communications Architectures & Protocols.* The issues discussed at this conference would have been research concepts and results of interest to those doing research and advanced development in the area of systems of networked computers.

27. The paper describes its context as follows: "In the UNIX operating system [9], each command is in fact a binary file that is loaded into memory and executed. In our model of distributed environment the user interacts with an executive on a personal workstation connected to a local network, and views commands not as binary files, but as *services* provided by *servers* in the network. The user no longer is logged in at a particular host in the network, but accesses the computing engine using the workstation to locate and execute services on its behalf." (Korb at p. 68, col. 2, para. 2.)

28. Korb's "services" or "binary files" are computer programs installed and executing on the servers, not the personal workstations, so do not represent functionality that the '078's controller would be "programmed to perform." Korb requires that the personal workstations and servers communicate with each other over a local area network. The '078 patent does not teach or imply communications with its controllers over a local area network, but rather over a dial-up telephone wide area network. So, most of Korb's context does not apply to the '078 patent's requirements, and Korb's teachings certainly do not directly apply to the '078 patent.

29. Korb is concerned with the issue of how to support a user on a personal workstation who wishes to perform a computation that chains together services already installed on different servers such that the results computed by one service are to be used as input to another service. At the time the way this was done was to send the intermediate results to the personal workstation, which would then issue them to the next server in the chain. The personal

workstation was a performance bottleneck in this process. Korb proposes a mechanism for sending the intermediate results directly to the next server without involving the personal workstation. This problem, of course, is not at all relevant to the requirements of the '078 patent, nor to Zap2it.com's service.

30. There are only two statements in Korb that could possibly be misconstrued to relate to the issue at hand of downloading during the execution of an application program incremental functionality for it to perform. The first is: "The setup for a V pipeline involves downloading a local (to the workstation) helper program for each remote service to handle input and output for that service." (*Id.* at p. 72, col. 2. para. 5.) The second is: "After creation of all connections, the executive downloads a local V program to connect to the input port of the first service, and to listen on all the local TCP ports." (*Id.* at p. 72, col. 2. para. 6.) In these statements the term "pipeline" refers to the mechanism used to transfer the results of a service's computation to its intended destination. In both of these statements the reference to "downloading a local program" refers to the process that is used to setup the chained computation prior to performing the computation. It does not in any way imply downloading during the execution of the computation incremental functionality for it to perform.

31. While Mr. Cole has not specified what in particular in Korb supports his contention that it teaches downloading during the execution of an application program incremental functionality for it to perform, it is my opinion, for the reasons stated above, that Korb does not teach this, either explicitly or inherently.

32. The third, and final document proffered by Mr. Cole to support his contention that, as of 1992, "examples of systems that downloaded or installed incremental functionality while the application program was being executed were common" is the Bell document. At his

deposition, Mr. Cole identified the following passage in Bell, Exhibit F to his Supplemental Report, to support his opinion: The section entitled "Static and Dynamic Assignment of Programs to Nodes" on page 16. (Cole Deposition Transcript dated June 21, 2007 at p. 244-245) In my review of Bell, neither the passage identified by Mr. Cole nor any other portion of the article supports Mr. Cole's opinion that "At the time of the '078 patent's priority date, examples of systems that downloaded or installed incremental functionality while the application program was being executed were common." (Cole Supplemental at ¶37.)

33. The author of this document, C. Gordon Bell, was a prominent computer designer and developer at the time the paper was published (1986), credited as one of the inventors of the minicomputer, and one of the founders of the Digital Equipment Corporation (DEC), a prominent and successful developer and supplier of minicomputer systems. The document, entitled "Toward A History Of (Personal) Workstations (Draft)" was apparently prepared by the author for presentation as the keynote address at a technical conference sponsored by the ACM. However, I cannot ascertain that the document was indeed published, or the keynote speech given.

34. The context of the Bell paper is the same as that of the Korb document, namely a multiplicity of computers interconnected via a local area network such that computer programs (services) installed on one or another of the computers (servers) can be used to perform computations on behalf of programs residing and executing on a different server. Some of the servers provide a graphical user interface to users, and are called by Bell "personal workstations." Such systems are called, generically, distributed computer systems. As with Korb's context, Bell's context is very different from, and therefore not relevant to, the '078 patent or Zap2it.com's service.

35. One small variation in Bell from the context of Korb is the concept of the "file server." About this concept Bell says: "LANs differ from Wide Area Networks (WANs) in that they assume a low latency, high bandwidth interconnect. This permits file access as well as file transfer applications. With file access, it is possible to remotely locate part or all of a system's mass storage to a file-serving computer. File access requires bandwidth and latency which are roughly equal to that of a disk (i.e., 10 Mbs rates); file transfer can be done at substantially slower rates (56Kbs to 1 Mbs)." In a distributed computer system that contains such a file server, the application programs can all be "installed" to the file server, where they reside until instantiated (initiated). At instantiation, or during execution, the actual program files in binary digit form will be copied to the local memory of the server assigned to execute the program, just as these files are copied to the local memory of a personal computer from its hard drive when the program is executed on it.

36. The only statements in Bell identified by Mr. Cole to support his assertion that it shows an example of a system that downloaded or installed incremental functionality while the application program was being executed are the following: "Ideally, a user can decide on how to compute on a completely variable basis at the following times:

- at system purchase or rent time ...
- at system use time ... here work is bound statically to a particular set of system resources. Most likely, particular nodes would execute special programs on data located at the node.
- at task time on the basis of reliability. VAX clusters provide for dynamic allocation of work among the computers in the cluster to affect load balancing since files are centralized.
- at task-use time on a completely dynamic basis, ranging from computing on his own local system to be able to select any resource and move work dynamically while programs are in execution. With this ability, as a program goes through its

various stages of development, it might be moved from small system to large system to take advantage of increased computational power at higher level nodes.

- at task time on a dynamic basis with the ability to acquire arbitrary resources to engage in parallel computation."

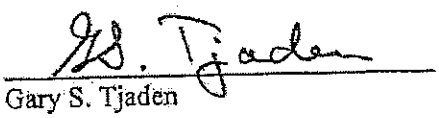
37. However, these statements teach something different than what Mr. Cole asserts. The statements refer to a computer system in which all of the programs are already "installed" on whatever servers they will be executed before the user begins the computation of interest. The entire collection of computers together with the local area network that interconnects them is a *single system*. During execution of the user's computation the data may be dynamically reassigned to another server where the appropriate program is installed, as taught also by Korb, but in no sense is incremental functionality downloaded or installed while the user's computation is being executed. In fact, the computers to which Bell refers, called VAX, that were manufactured by the company he founded, are the same computers used by Korb in its experiments.

38. In summary, for the reasons detailed above, the three documents cited by Mr. Cole in his Cole Supplemental Report, Goldszmidt, Korb and Bell, do not show that "at the time of the '078 patent's priority date, examples of systems that downloaded or installed incremental functionality while the application program was being executed were common." Rather, if anything they show that such functionality was not at the time commonly known to or understood by a skilled artisan, and certainly not used by consumer personal computers in user's homes.

39. I understand that TV Guide stated in its claim construction brief: "Indeed, it was common in the early 1990's, as it is today, for programs to be placed on a computer connected to a network as the computer is being used. (See Cole Ex. B at ¶¶ 121-123; Cole Ex. C at ¶ 37, Exs

D, E and F)," and I understand that Cole Ex. B refers to Cole's Rebuttal Report on Invalidity and Cole Ex. C refers to the Cole Supplemental Report. This statement mischaracterizes Mr. Cole's statements in the Cole Rebuttal Report on Invalidity and Cole Supplemental, and my statements in my Non-Infringement Rebuttal Report to which Cole was responding. I find nothing in Mr. Cole's reports to support this statement. This statement does not bear at all on the rationale for TMS's proposed construction of the term **"controller being programmed to perform,"** found in Claim 8 of the '078 Patent, to mean "computer application programs are installed on the controller prior to its use by a user to perform". This statement also is incorrect. It was not common in the early 1990's for programs to be placed, or installed, on a computer connected to a network as the computer was being used. I reserve the right to respond in the event Mr. Cole provides support for TV Guide's statement that "Indeed, it was common in the early 1990's, as it is today, for programs to be placed on a computer connected to a network as the computer is being used."

I declare the foregoing to be true and correct under penalty of perjury under the laws of the United States, and that this Declaration was executed on November 20, 2007, in Alpharetta, Georgia.

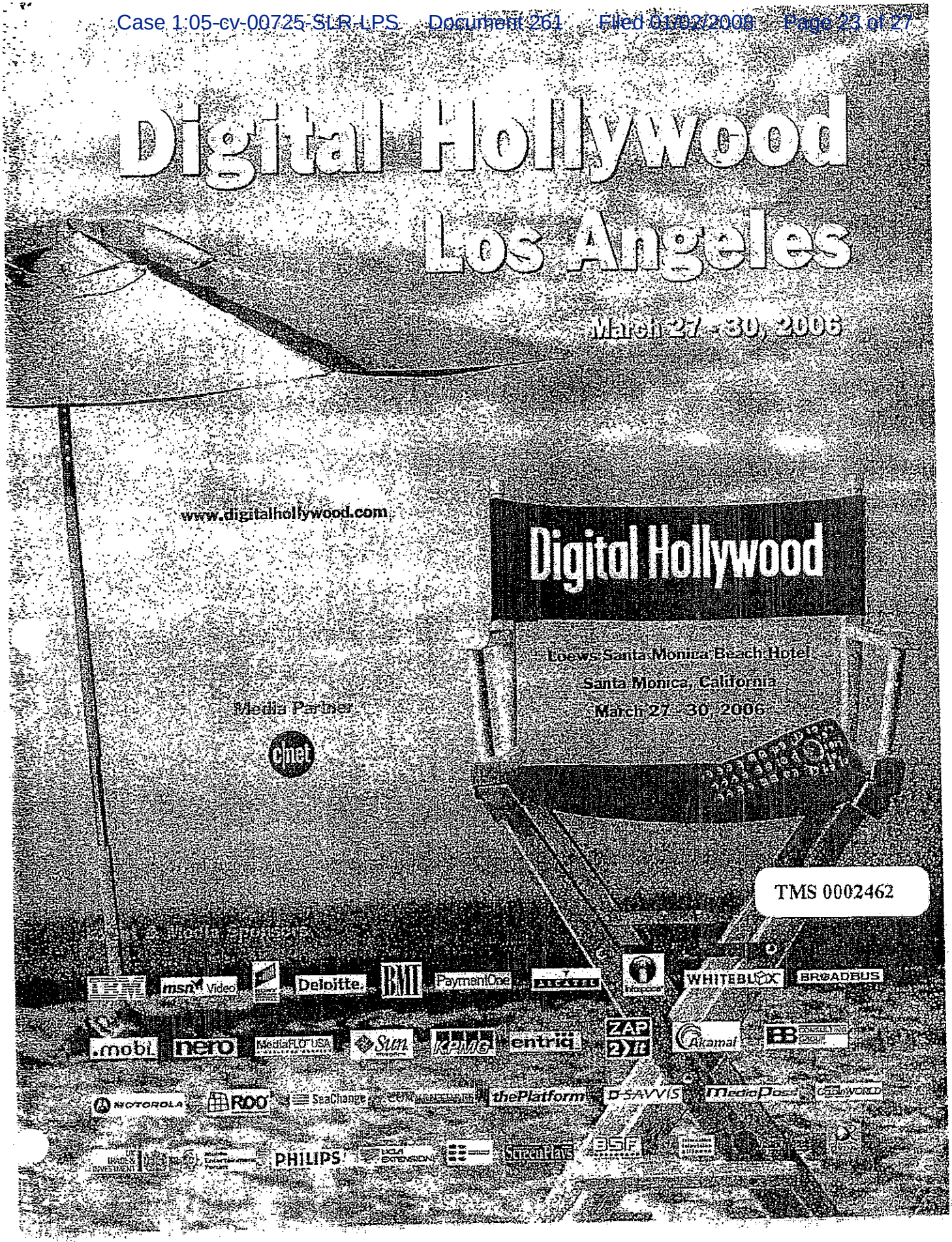


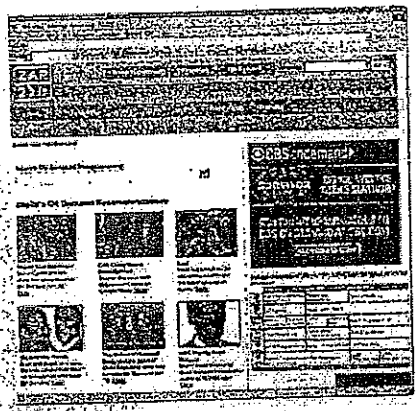
Gary S. Tjaden

EXHIBIT D

REDACTED IN ITS ENTIRETY

EXHIBIT E


Digital Hollywood
Los Angeles
March 27 - 30, 2006
www.digitalhollywood.com
Digital Hollywood
Loews Santa Monica Beach Hotel
Santa Monica, California
March 27 - 30, 2006
Media Partner
cnet
TMS 0002462
IBM
msn Video
Deloitte.
BMI
PaymentOne
WHITEBUX
BROADBUS
.mobi
nero
MediaFLO USA
Sun
KPMG
entriq
ZAP 2 it
Akamai
MOTOROLA
RDO
SeaChange
thePlatform
D-SAVVIS
MediaDoor
PHILIPS
ScreenRain
BSF



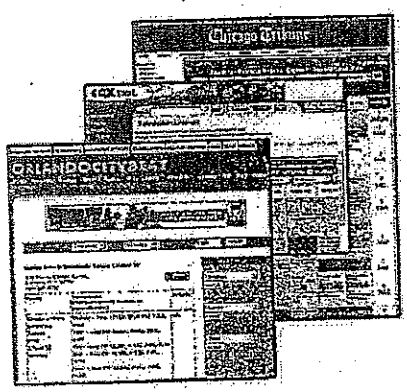
Zap2it's new On-Demand content area provides users with an easy-to-use guide to the evolving world of entertainment choices. We're connecting users with content anywhere and everywhere there's something worth watching.



Zap2it is an effective vehicle for marketers to drive tune-in, box office and DVD sales. Our On-Demand content offers new advertising and sponsorship opportunities.

We reach over 3 million unique visitors per month and our audience is tracked by @Plan. Contact us for a demographic profile or a customized Advertiser Planning report.

source: Nielsen//NetRatings



Our affiliate network includes hundreds of sites that run **Zap2it**-branded TV grids and movie showtimes. Zap2it's affiliate advertising network includes key listings pages in the entertainment areas of leading local newspaper websites.

TMS 0002463

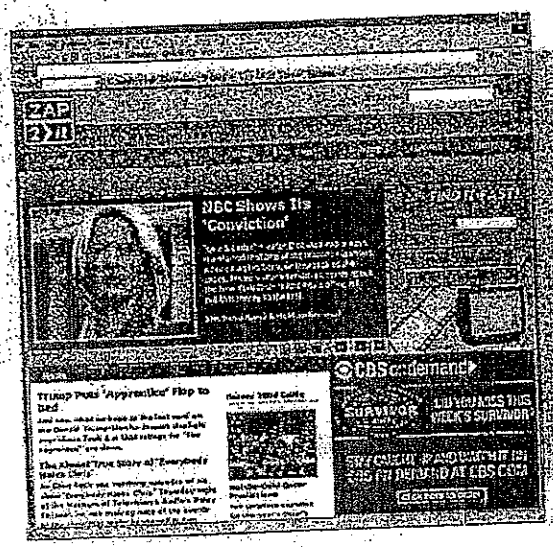
ZAP 2 it

What to Watch. Where to Watch It

http://www.zap2it.com

Tribune Media Services data powers many of the world's leading portals and entertainment company web sites. TMS is a leading provider of print, online and on-screen entertainment listings and editorial information to newspapers and online media, cable and satellite operators and consumer electronics companies. Our advertising networks provide opportunities to reach nearly 70 million consumers through Zap2it.com and major market newspaper supplements. TMS is a division of Tribune Company.

We help viewers decide what to watch. And where to watch it.



The new era of on-demand and on-the-go entertainment brings more choices than ever before.

Cable VoD. Movie download services. Video-rich web sites. TV-on-Mobile.

And a continuous stream of new announcements about new ways to enjoy movies and TV.

As the leader in entertainment information and advertising networks, Tribune Media Services knows how important it is to help consumers understand the new choices.

And TMS' Zap2it is in the forefront of that change.

More than 3 million consumers a month turn to Zap2it.com for the latest movie and TV information.

We're bringing content providers and audiences together everywhere there's something worth watching.

TMS 0002464

ZAP 2IT.

What to Watch. Where to Watch It

http://www.zap2it.com

Contact Information:
Advertising Sales
Rick Gables – West Coast: rgables@tribune.com; 310.452.6021
Marc White – East Coast: mrwhite@tribune.com; 212.210.2766

Business Development
John Zelenka: jzelenka@tribune.com; 312.222.4871

TRIBUNE MEDIA SERVICES

EXHIBITS F - K

REDACTED IN THEIR ENTIRETY

IN THE UNITED STATES DISTRICT COURT
DISTRICT OF DELAWARE

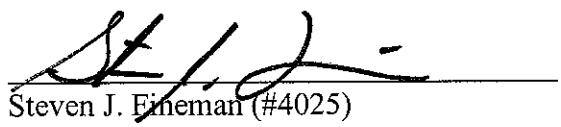
CERTIFICATE OF SERVICE

I hereby certify that on January 2, 2008, I electronically filed the foregoing document with the Clerk of Court using CM/ECF which will send notification of such filing(s) and Hand Delivered to the following:

Richard K. Herrmann, Esquire
Mary B. Matterer, Esquire
Morris James, LLP
500 Delaware Avenue
Suite 1500
Wilmington, DE 19801-1494

I hereby certify that on January 2, 2008, I have sent by electronic mail, the foregoing document to the following non-registered participant:

Mark A. Pals, P.C., Esquire
Kirkland & Ellis, LLP
200 East Randolph Drive
Chicago, IL 60601



Steven J. Fineman (#4025)
Richards, Layton & Finger, P.A.
One Rodney Square
P.O. Box 551
Wilmington, Delaware 19899
(302) 651-7700
fineman@rlf.com