(stating "Milestone Review ... basic HW/SW bootup on 1st proto – Sep ... DSP playing sounds ... Late Basic MP3 playback (RAM) ... (disk) ... Rev 2 Proto, in form factor ... Nov"). The reference to "Rev 2 Proto, in form factor" refers to a second prototype that would later be implemented having the size and shape as desired for the ultimate Nomad Jukebox product. *Id.* The earlier prototypes were not "form factor," meaning they were larger and square-shaped and not designed to actually fit in the plastic casing that was being designed. *See, e.g.*, Exh. T.

38. At the Oasis Engineering Meeting on October 13, 1999, I reported on certain issues concerning the embedded software. *See* Exh. W, p. 2. My co-inventors on the '433 Patent, David Bristow and Ron Goodman, were also present at the Oasis Engineering Meeting on October 13, 1999, we talked about efforts in "storyboarding" the user interface. *Id.*, p. 3.

39. At another Oasis Engineering Meeting that took place some time after Oct. 20, 1999, but before Dec. 9, 1999, one of my colleagues gave a status report on the embedded software development, reporting that: "Howard [Egan] has been working on file system ... The basics work ... Load, save, create files"; "Playback mgr, DSP Mgr. ... MP3 decode works in Playback mgr"; and "Ron's UI stuff integrated into Oasis." *See* Exh. W, p. 10.

40. At a subsequent meeting that took place some time after Dec. 2, 1999, but before Dec. 9, 1999, Mr. Freeman reported on the status of the Oasis hardware development, indicating that: "Oasis PCB in assembly"; "test fixture in fab Thu 12/2/99"; "DVT plan being written"; "Prog guide/Theory of Ops doc being written." *See* Exh. W, p. 12. The notes also indicate that I was present at this meeting, and that I said "MP3 decode from disk works." *Id.* These notes indicate to me that in the first week of December, 1999, I received a form-factor prototype that I could use to begin testing the Oasis operating system in firmware on a hardware prototype, as opposed to by using an emulator. *Id.* The notes also say that I wanted Scott W. "commit to what can be delivered for CES." *Id.* The notation "CES" refers to the upcoming Consumer Electronics Show 2000 ("CES 2000"), which was scheduled to take place in Las Vegas, NV on Jan. 5 through Jan. 10, 2000. The Oasis R&D team had already been notified some time in September of 1999 that Creative was planning to introduce the Nomad Jukebox at CES 2000. It was understood by all members of the Oasis R&D team that this meant we needed to have a robust working prototype fully tested and ready to present at least a day before the show was scheduled to begin on Jan. 5, 2000. So, the target date was set for Jan. 4, 2000.

41. At a meeting on Dec. 9, 1999, I gave a status update on the development of the Oasis embedded software. *See* Exh. W, pp. 14-15. Prior to that day, the hardware sub-group had provided

me with a functional hardware prototype having all the same components shown in Exh. Q. I know this based on my memory of the events, and also because Mr. Freeman's notes of Dec. 9, 1999 indicate that I talked about the need to integrate "boot loader" software into the system. *See* Exh. W, p. 15 (stating "Intertrust secure Bootloader – we have to write bootloader. Intertrust portion in read-only part of the Boot ROM").

42. Using the prototype Nomad Jukebox provided to me prior to December 9, 1999, I had begun testing the Oasis operating system. *See, e.g.* Exh. W, p. 15 (stating "UI manager task running, button manager ... Library traversal system in place ...") Prior to Dec. 9, 1999, I had begun a testing process wherein I compiled the Oasis source code, and executed it as firmware on the ARM processor of the prototype. *See* Exh. Q. In conjunction with testing the Oasis source code, I used a personal computer (PC) connected to an Oasis Test Fixture Board via a standard Joint Test Action group ("JTAG") adaptor, wherein the prototype Nomad Jukebox connected directly to the Oasis Test Fixture Board. Exhibit U shows photographs taken recently of the Oasis Test Fixture Board I began using prior to Dec. 9, 1999 for purposes of testing the Oasis operating system. *Id.* I note the Oasis Test Fixture Board bears a copyright date of 1999. *See* Exhibit U. This test board was designed by Mr. Freeman and others specifically for the purpose of testing the source code for the embedded software, including the Oasis operating system. *See, e.g.* Exh. W, p. 12 (stating "Test fixture in fab Thu 12/2/99 ... DVT plan being written"). Mr. Freeman's notes concerning the "Test fixture" refer to the Oasis Test Fixture Board shown in exhibit U. *See, e.g.* Exh. W, p. 12; *see also* Exh. U. The notation "DVT plan being written" refers to a plan for a device verification test ("DVT") for verifying the operation of the prototype Nomad Jukebox. *See, e.g.* Exh. W, p. 12; *see also* Exh. U.

43. Prior to Dec. 9, 1999, I began testing the Oasis operating system by compiling the Oasis source code, and downloading . the compiled code via the JTAG adaptor to the ARM processor of the prototype. *See* Exh. Q, Exh. U. The Oasis block diagram of exhibit Q shows the basic components of the prototype. As mentioned above, Mr. Freeman's notes of Dec. 9, 1999 indicate that I talked about the need to integrate "boot loader" software into the prototype. *See* Exh. W, p. 15 (stating "Intertrust secure Bootloader – we have to write bootloader. Intertrust portion in read-only part of the Boot ROM"). Referring to the diagram of exhibit Q, the Nomad Jukebox prototype design called for "bootloader" code to be stored in the "Boot Flash" memory. *See* Exh. Q. The purpose of the bootloader can be described as follows. Upon initial power-up of the system, the ARM processor would access and execute the bootloader code from the boot flash memory, which would then load the Oasis operating system from the Hard Disk to the dynamic random access memory ("DRAM") so that the ARM processor could then begin executing the Oasis operating system to control all

embedded software functions on the portable media player. *See* Exh. Q. However, as of Dec. 9, 1999, the bootloader code was not yet written and installed into the boot flash memory. *See* Exh. Q; Exh. W, p. 15. In order to proceed with the testing without the boot loader, upon power-up of the prototype, I configured the PC to control the ARM processor on the prototype via the JTAG adaptor so that the ARM processor loaded the Oasis operating system directly from the hard disk drive to the DRAM. This was a simple by-pass solution that enabled me to fully test the Oasis operating system, including all of the functions described above as practicing claims 1-33 of the `433 Patent.

44. Using the techniques described in the previous paragraph, I was able to power-up and initialize the Nomad Jukebox prototype without having a bootloader, and thus I was able to test the compiled Oasis operating system by executing it directly on the prototype without the use of any software emulation. *See,* Exh. W, pp. 14-15; *see also* Exh. U. Mr. Freeman's notes of Dec. 9, 1999 also state "UI Manager task running, button manager ... library traversal system in place." *See,* Exh. W, pp. 14-15. This indicates to me that prior to Dec. 9, 1999, I had been able to power-up and test parts of the Oasis operating system on the Nomad Jukebox prototype without having a bootloader. *Id.* These notes are in accordance with my memory of the events during the first few weeks of January, 1999, which I remember very well because I was very excited and determined to meet the goal of publicly demonstrating a Nomad Jukebox prototype at CES 2000, which was at that time less than a month away.

45. Mr. Freeman's notes on the report I gave on Dec. 9, 1999 also state "want 'limping along' unit that can play songs next week (old boards)." *See,* Exh. W, pp. 14-15. I coined the term "limping along" unit on or about the time of the Dec. 9, 1999 meeting to refer to a stand-alone Nomad Jukebox prototype detached from the Oasis Test Fixture Board and other test equipment, that would could properly execute the Oasis operating system. *See* Exhs. AA through OO. Once again, our goal in December of 1999 was to publicly demonstrate a Nomad Jukebox prototype at CES 2000, which was at that time less than a month away. I called it a "limping along" unit sort of as a term of endearment because the Oasis R&D was well aware that any early prototype is likely to have sporadic bugs or glitches. However, we all took very serious the prospect of demonstrating a prototype at CES 2000, which is the largest and most important annual consumer electronics show in North America. As such, we knew that the stakes were extremely high because it would have been a major embarrassment if any of the Nomad Jukebox prototypes presented at CES 2000 were to "crash."

46. During a meeting on Dec. 15, 1999, I gave another update on development of the Oasis embedded software. *See* Exh. W, pp. 16-17. Mr. Freeman's notes indicate "'Limping along' unit

could be ready later this week? Then will be made more robust so can survive demo at CES." *Id.*, p.17. This means that the bootloader code was still not installed.

47. Based on my memory, my inspection of the CVS check-in logs (*see* Exh. Z), my inspection of the Dec. 14, 1999 Oasis source code (*see* Exhs. A through O), and also based on Mr. Freeman's notes on Dec. 9, 1999 (*see* Exh. W at pp. 14-15), I am sure that at least as of Dec. 14, 1999, I had fully tested the Oasis operating system – which implemented all of claims 1-33 of the `433 Patent - as actual firmware running on a Nomad Jukebox prototype. S*ee* Exh. W, pp. 14-15 (stating "UI Manager task running, button manager ... library traversal system in place.") It is true that as of Dec. 9, 1999, many aspects of the prototype needed to be de-bugged, and it is also true that the Oasis operating system needed to be loaded by JTAG control on power-up due to the absence of a bootloader. However, the bootloader was <u>not</u> essential to demonstrating that the Oasis operating system (including <u>all</u> of the functionality described above in Table A) was working for its intended purpose. Therefore, as of Dec. 14, 1999, I had fully tested all of the functions correlated in the above table with claims 1-33 of the `433 Patent, and these functions were working for their intended purpose.

48. Prior to Dec. 14, 1999, before I began testing the Oasis operating system as compiled firmware on a Nomad Jukebox prototype (as the notes show I was doing throughout the first few weeks of December of 1999), I had previously tested the Oasis operating system (including <u>all</u> of the functionality described above in Table A) using a software emulator. For reasons that will be understood by anyone skilled in the development of firmware in the 1999 time period, I would not have begun testing the firmware without first testing the source code in an emulator environment, where it was much easier to debug.

**<u>Diligence Towards A Second Reduction to Practice on January 4, 2000</u>**

49. As explained above, I believe that the Dec. 15, 1999 Oasis source code demonstrates complete conception of claims 1-33 of the `433 Patent (*see* Table A, above) at least as early as Dec. 15, 1999. I also believe that the above-described emulation testing and/or the firmware testing was sufficient to demonstrate that the inventions recited in claims 1-33 of the `433 Patent were working for the intended purpose.

50. Alternatively, I submit that: (1) the Jan. 4, 2000 Nomad Jukebox prototypes demonstrated at CES 2000 constituted an actual reduction to practice of the inventions of claims 1-33 of the `433 Patent; (2) the Oasis R&D team was diligent in its effort to reduce to practice the inventions of claims

1-33 starting on Dec. 15, 1999 and proceeding on a daily basis up until the time of the actual reduction to practice on Jan. 4, 2000.

51. As explained above, during our development of the Nomad Jukebox in December of 1999, we were working toward a very aggressive internal target date of Jan. 4, 2000 for having a fully functional Nomad Jukebox ready to present at CES 2000. And as explained below, we were very successful in achieving that goal.

52. After the Oasis Engineering Meeting on Dec. 15, 1999, the Oasis R&D team including Mr. Freeman began working so hard and fast toward the CES 200 target date that Mr. Freeman largely stopped taking meaningful notes in his notebook in the time period between Dec. 15, 1990 and January 11, 2000. *See* Exh. W, pp. 16-22. However, I have located numerous documents that describe the work that was done on reducing the inventions of the `433 Patent to practice during the time period between Dec. 15, 1999 and Jan. 4, 2000.

53. Exhibit Z is a copy of a Log of all CVS check-in activity for the Oasis source code between Nov.r 10, 1999 and Jan. 10, 2000. This log indicates that portions of Oasis source code were checked in by various members of the embedded software development team on the following dates: **12/14/99** (*see, e.g.,* Exh. Z, p. 4 showing a check-in by Howard Egan with notes indicating "Resynchronize after adding button scanner, transfer agent, now playing manager, LibTreeManager etc.); **12/16/99** (*see, e.g.,* Exh. Z, p. 196 showing a check-in by Andrei Veltchev with notes indicating "delay inserted to accommodate the D12's timing requirements"); **12/17/99** (*see, e.g.,* Exh. Z, p. 4 showing a check-in by Howard Egan with notes indicating "First cut with playback manager hooked to transfer agent Plays audio from disk"); **12/18/99** (*see, e.g.,* Exh. Z, p. 47 showing a check-in by Andrei Veltchev with notes indicating "added encoder event handling to the LCD manager"); **12/20/99** (*see, e.g.,* Exh. Z, p. 11 showing a check-in by Howard Egan with notes indicating "Miscellania"); **12/22/99** (*see, e.g.,* Exh. Z, p. 70 showing a check-in by Howard Egan with notes indicating "Moved from sysmgr"); **12/23/99** (*see, e.g.,* Exh. Z, p. 75 showing a check-in by Howard Egan with notes indicating "Latest boot loader modifications and inclusion of some new hard coded music."); **12/24/99** (*see, e.g.,* Exh. Z, p. 11 showing a check-in by Gerald I. with notes indicating "Fixed some transport issues"); **12/26/99** (*see, e.g.,* Exh. Z, p. 10 showing a check-in by Howard Egan with notes indicating "Fixed several transport key issues, and changed fonts and LCD Display size to 6 lines (total)"); **12/27/99** (*see, e.g.,* Exh. Z, p. 10 showing a check-in by Howard Egan with notes indicating "Added status line class to LCDPage for use by both now playing screen and qlist screen. Added elapsed playing time calculations and functions to dsp manager for use by status line."); **12/28/99** (*see, e.g.,* Exh. Z, p. 35

showing a check-in by Howard Egan with notes indicating "Miscellaneous bug fixes"); **12/29/99** (*see, e.g.*, Exh. Z, p. 35 showing a check-in by Howard Egan with notes indicating "Added fix to filesystem to take an initial file size on creation. "Cause double indirect is broken currently""); **12/30/99** (*see, e.g.*, Exh. Z, p. 35 showing a check-in by Howard Egan with notes indicating "Better fonts, cosmetics"); **12/31/99** (*see, e.g.*, Exh. Z, p. 5 showing a check-in by Howard Egan with notes indicating "Added INode debug output statements."); **01/01/2000** (*see, e.g.*, Exh. Z, p. 5 showing a check-in by Howard Egan with notes indicating "Added system menu, including dsp effects control"); **01/03/2000** (*see, e.g.*, Exh. Z, p. 15 showing a check-in by Howard Egan with notes indicating "CES Release"); and **01/04/2000** (*see, e.g.*, Exh. Z, p. 21 showing a check-in by Howard Egan with notes indicating "CES Final").

54. As noted above, the CVS Log indicates that members of the Oasis R&D team checked in source code revisions on every day between Dec. 14, 1999 and Jan. 4, 2000, except the following days: Dec. 15, 1999; Dec. 19, 1999; Dec. 21, 1999; Christmas day, 1999; and Jan. 2, 2000. However, I am absolutely sure that most of the members of the members of the Oasis R&D team worked diligently every single day between Dec. 14, 1999 and Jan. 4, 2000, with the possible exception of Christmas day, 1999. I worked on revising and testing the Oasis operating system on every single day in this time period, except Christmas day. In addition, I should explain that each one of the logged source code revisions may reflect more than one day's worth of work leading up to the check-in date. Moreover, the fact that no check-ins on a given date certainly does not mean that there was no embedded software development activity on that date because the Oasis source code developers followed the standard practice of testing all source code revisions before checking them in, and this was often a complicated and time consuming process. I note that there was a very well attended Oasis Engineering Meeting on Dec. 15, 1999. *See* Exh. W, pp. 16-18. Mr. Freeman's notebook shows that at least he was working on development of the Nomad Jukebox on Dec. 19, 1999. *See* Exh. W, pp. 18-19. I am certain I worked that day as well. Thus, the only days in the period for which I cannot find direct evidence of work activity are: Dec. 21, 1999; Jan. 2, 2000 (which was a Sunday); and Christmas day, 1999. However, as I said, I am certain that I personally worked on every day between Dec. 14, 1999 and Jan. 4, 2000, except Christmas day. In this time period, I estimate that I worked about 80 hours per week on further development of the Nomad Jukebox embedded source code. And I am sure that the other members of the Oasis R&D team worked at a similar pace in developing various other components of the Nomad Jukebox. As explained, the reason for this tireless pace was that we had promised to deliver a working Nomad Juekbox prototype in time to demonstrate it at the CES 2000 show in Las Vegas, NV starting on January 5, 2000.

55. An e-mail sent by Dan Freeman on Dec. 28, 1999 to Andrei Veltchev and Howard Egan regarding "ARM Initialization Pwr Up/Down for CES" shows that Dan Freeman had been working various aspects of the hardware, and his message indicates that he had solved the bootloader problem, which I had previously discussed at the Oasis Engineering Meeting on Dec. 9, 1999. *See* Exh. X; *see also* Exh. W, pp. 15-16.

56. Dan Freeman sent another e-mail on Dec. 30, 1999 to a number of people including myself. *See* Exh. Y. This email attached a Word document, entitled "Script Guideline: Public Demonstration." *Id.* The purpose of the document was to train Creative sales and marketing personnel on how to operate the Nomad Jukebox portable media players that would be ready to demonstrate at CES 2000. *Id.* This document provides direct evidence about what functions were ready for the "demonstration" at CES 2000. Id. In fact, the document provide directions for how to perform the demonstration, including the following excerpts:

- "Lets start with the LibraryKey – show that this **skips between the top library Screen and the 'now playing' Screen**"
- "SoftKeys – repeat that the **softkeys are labeled depending on the screen**, and for example, **when searching the library, they are used to expand the view via different categories**";
- "Show the use of the ScrollKeys by moving up and down **lists of albums, styles, artists or tracks**";
- "Mention 6GB HD holds 100 hours of CD-quality music ...";
- "Touch the LibraryKey again to return to the **top Library screen** and **select an album**. (let the audience **choose a style**). Show how to **play this directly from the Library by pressing the PLAY** transport key"
- "You find and **select** what you want to hear and simply **press Play**'"
- "Show that the Jukebox automatically displays the **"Now Playing" screen** soon after audio has started ...see the list of tracks waiting to be played";
- "Now direct the audience to the sound quality. Have them put on a headset (if available)";
- "Remember: if **the PlayKey is pressed** when the Library Screen is active, it will **immediately play the selection** and will not act as pause"
- "**Building and playing playlists** is an important part of the JukeBox's function. We've shown how easy it is to **immediately Play any Album from the library by selecting and using the PlayKey**. However, whenever you repeat this action with the PlayKey, whatever is playing will stop and the new selection will take its place on the "virtual turntable". Show an example of this by selecting and playing a track from the Library followed by a second "played" track after about 10 seconds.
- "**To build up a list of selections**, instead of pressing the PlayKey when searching through the library, **press the QueueKey** ... This **will add your selection to the currently playing list**, and you can check this by looking at the

list view of the PlayScreens. After any listening session, **you can choose to save all the music you have been listening to as a playlist** for future occasions."

*See* Exh. Y, p. 3 (emphasis added).

57. As indicated in the CVS log, I personally checked in the Oasis source code that was used in the Nomad Jukebox prototypes presented at CES 2000 in accordance with the above-cited demonstration instructions. *See* Exh. Z, p. 21 (showing a check-in on 01/04/2000 by Howard Egan with notes indicating "CES Final").

58. According to schedule, the NOMAD® Jukebox was indeed presented to attendees at the Consumer Electronics Show in Las Vegas, NV, which began on January 5, 2000. *See* Exh. PP, Exh. QQ. I was present at the show, and I participated in demonstrating the NOMAD® Jukebox. During and after the CES 2000 show, the NOMAD® Jukebox received rave reviews. *See* Exh. PP, Exh. QQ.

59. Presented below in Table B is a chart explaining how the Jan. 4, 2000 Oasis source code (*see* Exhs. AA through OO) - which was embedded as fully operative firmware in the NOMAD® Jukebox portable media players presented at CES 2000 - implemented each and every element of claims 1-33 of the '433 Patent.

**Table B**

| Claim Elements | Jan. 4, 2000 "CES Final" Oasis Source Code |
|---|---|
| 1. A method of selecting at least one track from a plurality of tracks stored in a computer-readable medium of a portable media player configured to present sequentially a first, second, and third display screen on the display of the media player, | The Nomad Jukebox "Script Guideline : Public Demonstration" explains many features of the Nomad Jukebox portable audio players that were demonstrated at CES 2000. The Nomad Jukebox portable audio players included: a liquid crystal display screen (*see* Exh. Y at pp. 3-4); user control buttons such as ScrollKeys and SoftKeys (*see id.* at p. 3); a hard disk drive for storing songs (or tracks) (*see id.* at pp. 3 and 7.); electronics for playing music through headphones or a speaker (*see id.* at p. 3.).<br><br>The Oasis source code (*see* Exhs. AA through OO) was embedded as fully operative firmware in the NOMAD® Jukebox portable media players presented at CES 2000.<br><br>On or before Jan. 4, 2000, the prototype portable media player was tested. The test results demonstrated that the prototype successfully implemented all of the functions of the Jan. 4, 2000 version of the Nomad Jukebox source code.<br><br>The source code files LibraryTree.cpp and LibraryTree.h implement a hierarchy in the form of a library tree. *See* Exhs. BB and CC. The library tree stores human-readable information that is shown in a user interface, and also stores the actual details of where to find a particular |

| Claim Elements | Jan. 4, 2000 "CES Final" Oasis Source Code |
|---|---|
| | track in the file system. *Id.* A display screen is displayed for every level of the hierarchy. *Id.* Thus, the entire display is replicated by the library tree. *Id.* This way the user-interface code traverses the tree and displays any items(s) at the current node. *Id.*<br><br>A plurality of musical "tracks" are stored in a computer-readable medium accessed via the library tree. *See* file LibraryTree.cpp at Exh. BB, p. 2 (referring to a "tracks directory"); *see also* Exh. CC.<br><br>As indicated by developer notes in the source code, the "Class CLibTreeMgr creates and makes modifications as necessary to a tree structure whose purpose is to make the track data on disk logically and quickly navigable. It is currently rebuilt every time the system boots." *See* Exh. BB; Exh. CC.<br><br>On entry into the Library display screen, the elements of the top level of the Library Tree are displayed by the CLCDLibPage class. *See* Exhs. DD and EE. The top level of the Library Tree has the following categories: "**Albums**," "**Artists**," "**Styles**," "**Play Lists**." *See, e.g.,* Exh. BB, pp. 1-2.<br><br>The display of the above elements is one example of a "first ... display screen." From here the user has the option of opening any of the displayed categories by pressing an "Open" softkey (*see* CLCDLib Page::Softkey1Handler() at Exh. DD, pp. 1 and 3; Exh. EE, p. 1).<br><br>The user moves the selection up and down lists by pressing up and down arrow keys, wherein the key presses are handled by the CLCDMgr (*see* Exh. FF, p. 5) which in turn tells the CLCDLibPage (*see* Exhs. DD and EE) to scroll the highlight up or down.<br><br>When the user presses "Open" with the "Albums" item highlighted, the CLCDLibPage class (*see* Exhs. DD and EE) displays a current list of Albums, which is one example of a "second ... display screen." As explained above, the user can scroll up and down the list, and either Open, Queue, or Play an item. At this level, the Close softkey is also active. (*See* Exhs. DD and EE.)<br><br>When the user again presses "Open" with a selected album highlighted, the CLCDLibPage class (*see* Exhs. DD and EE) redraws the screen showing a list of tracks corresponding to the album. This is one example of a "third display screen." At this point CLCDLibPage (*see* Exhs. DD and EE) again changes the soft key labels to replace "Open" with "Details". Pressing a Details button for any selected track will display more detailed information, such as album, artist, duration. |
| the plurality of tracks accessed according to a hierarchy, | The source code files LibraryTree.cpp and LibraryTree.h implement a "hierarchy" in the form of a library tree. *See* Exhs. B and C. The library tree is built at system startup by traversing tracks stored in a file system. *Id.* Additionally, as the user adds tracks, the library tree is expanded. *Id.* The library tree stores human-readable information |

| Claim Elements | Jan. 4, 2000 "CES Final" Oasis Source Code |
|---|---|
| | that is shown in a user interface, and also stores the actual details of where to find a particular track in the file system. *Id.* A display screen is displayed for every level of the hierarchy. *Id.* Thus, the entire display is replicated by the library tree. *Id.* This way the user-interface code traverses the tree and displays any items(s) at the current node. *Id.* The library tree is implemented in class by the files LibraryTree.cpp (*see* Exh. BB) and LibraryTree.h (*see* Exh. CC).<br><br>Tracks are "accessed" using the "hierarchy" in the form of a library tree. *See* Exhs. BB and CC.<br><br>As indicated by developer notes in the files LibraryTree.cpp and LibraryTree.h, "Class CLibTreeMgr creates and makes modifications as necessary to a tree structure whose purpose is to make the track data on disk logically and quickly navigable. It is currently rebuilt every time the system boots." *See* Exh. BB, p. 1; Exh. CC, p. 1. |
| the hierarchy having a plurality of categories, subcategories, and items respectively in a first, second, and third level of the hierarchy, the method comprising: | The source code files LibraryTree.cpp and LibraryTree.h implement a hierarchy in the form of a library tree. *See* Exhs. B and C.<br><br>The library tree stores human-readable information that is shown in a user interface, and also stores the actual details of where to find a particular track in the file system. *Id.* A display screen is displayed for every level of the hierarchy. *Id.* Thus, the entire display is replicated by the library tree. *Id.* This way the user-interface code traverses the tree and displays any items(s) at the current node. *Id.*<br><br>The top level of the Library Tree (*see* LibraryTree.cpp ) displays the top level categories (*i.e.*, "**Albums**," "**Artists**," "**Styles**," "**Play Lists**") on a first display screen. *See, e.g.*, Exh. BB, pp. 1-2.<br><br>The categories are in a first "level" of the "hierarchy."<br><br>When the user presses "Open" with the "Artists" element highlighted, the CLCDLibPage class (*see* Exhs. DD and EE) gets a current list of Artists from an artist level of the library tree. This artist level is one example of a "second level" of the "hierarchy." The list of artists is one example of a plurality of "subcategories" in a "second level" of the "hierarchy."<br><br>Now if a user presses "Open" with a particular Artist item highlighted, the CLCDLibPage class (*see* Exhs. DD and EE) gets a current list of Albums for the selected Artist from an album level of the hierarchy. This album level is one example of a "third level" of the "hierarchy." The list of albums is one example of a plurality of "items" in a "third level" of the "hierarchy."<br><br>Now if a user presses "Open" with a particular Album item highlighted, the CLCDLibPage class (*see* Exhs. DD and EE) gets the tracks for the selected album from a track level of the hierarchy. This track level is one example of a "fourth level" of the hierarchy. This list of tracks is another example of a plurality of "items." |

| Claim Elements | Jan. 4, 2000 "CES Final" Oasis Source Code |
|---|---|
| | Highlighting a track and pressing the Play button cause the CLCDMgr (*see* Exh. FF and G) to call CNowPlayingMgr::Play() function (*see* Exhs. HH and II) to clear the contents of the NowPlayingQ, and replace it with the track and begin playing. |
| selecting a category in the first display screen of the portable media player; | On entry into the Library display screen, the elements of the top level of the Library Tree are displayed by the CLCDLibPage class. *See* Exhs. DD and EE.<br><br>The top level of the Library Tree (*see* LibraryTree.cpp ) displays the top level categories (*i.e.*, "**Albums**," "**Artists**," "**Styles**," "**Play Lists**") on a first display screen. *See, e.g.*, Exh. BB, pp. 1-2.<br><br>The display of the above categories is one example of a "first display screen." From here the user has the option of opening any of the displayed categories by pressing an "Open" softkey (*see* CLCDLib Page::Softkey1Handler() at Exh. DD, pp. 1 and 3; Exh. EE, p. 1).<br><br>The user moves the selection up and down lists by pressing up and down arrow keys, wherein the key presses are handled by the CLCDMgr (*see* Exh. FF, p. 5) which in turn tells the CLCDLibPage (*see* Exhs. DD and EE) to scroll the highlight up or down.<br><br>When the user presses "Open" with the "Albums" category highlighted in the "first display screen," the CLCDLibPage class (*see* Exhs. DD and EE) displays a current list of Albums. This is one example of "selecting a category in the first display screen of the portable media player." |
| displaying the subcategories belonging to the selected category in a listing presented in the second display screen; | When the user presses "Open" with the "Albums" category highlighted in the "first display screen," the CLCDLibPage class (*see* Exhs. DD and EE) displays a current list of Albums, which is one example of "subcategories belonging to the selected category in a listing presented in the second display screen."<br><br>As explained above, the user can scroll up and down the list, and either Open, Queue, or Play a selected one of the albums in the list. (*See* Exhs. DD and EE.) |
| selecting a subcategory in the second display screen; | When the user presses "Open" with the "Albums" category highlighted in the "first display screen," the CLCDLibPage class (*see* Exhs. DD and EE) displays a current list of Albums, which is one example of "subcategories belonging to the selected category in a listing presented in the second display screen."<br><br>The user can scroll up and down list of Albums, and activate either Open, Queue, or Play buttons to select one of the albums on the list. (*See* Exhs. DD and EE.) This allows for "selecting a subcategory in the second display screen." |
| displaying the items belonging to the selected | When the user presses "Open" with the "Albums" category highlighted in the "first display screen," the CLCDLibPage class (*see* |

| Claim Elements | Jan. 4, 2000 "CES Final" Oasis Source Code |
|---|---|
| subcategory in a listing presented in the third display screen; and | Exhs. DD and EE) displays a current list of Albums, which is one example of "subcategories belonging to the selected category in a listing presented in the second display screen."<br><br>The user can scroll up and down list of Albums, and activate either Open, Queue, or Play buttons to select one of the albums on the list. (*See* Exhs. DD and EE.) This allows for "selecting a subcategory in the second display screen."<br><br>When the user again presses "Open" with a selected album highlighted, the CLCDLibPage class (*see* Exhs. DD and EE) redraws the screen showing a list of tracks corresponding to the selected album. This is one example of "displaying the items belonging to the selected subcategory in a listing presented in the third display screen."<br><br>At this point CLCDLibPage (*see* Exhs. DD and EE) again changes the soft key labels to replace "Open" with "Details." Pressing a Details button for any selected track will display more detailed information, such as album, artist, duration. Pressing a Details button for any selected track will display more detailed information, such as album, artist, duration. |
| accessing at least one track based on a selection made in one of the display screens. | When the user again presses "Open" with a selected album highlighted, the CLCDLibPage class (*see* Exhs. DD and EE) redraws the screen showing a list of tracks corresponding to the selected album.<br><br>Highlighting a track and pressing the Play button causes the CLCDMgr (*see* Exh. FF and G) to call CNowPlayingMgr::Play() function (*see* Exhs. HH and II) to clear the contents of the NowPlayingQ, and replace it with the track and begin playing. This is one example of "accessing at least one track based on a selection made in one of the display screens." |
| 2. The method of selecting a track as recited in claim 1 wherein<br><br>the accessing at least one track comprises selecting a subcategory in the second display screen | The top level of the Library Tree (*see* LibraryTree.cpp ) displays the top level categories (*i.e.*, "**Albums**," "**Artists**," "**Styles**," "**Play Lists**") on a first display screen. *See, e.g.*, Exh. BB, pp. 1-2.<br><br>When the user presses "Open" with the "Albums" category highlighted in a first display screen, the CLCDLibPage class (*see* Exhs. DD and EE) displays a current list of Albums, which is one example of "a subcategory in the second display screen."<br><br>The user can scroll up and down the list of Albums, and activate either Open, Queue, or Play buttons to select one of the albums on the list. (*See* Exhs. DD and EE.) This allows for "selecting a subcategory in the second display screen."<br><br>The user moves the selection up and down the lists by pressing the up and down arrow keys. These key-presses are handled by the CLCDMgr (see Exh. FF, p. 5) who in turn tells the CLCDLibPage (see Exhs. DD and EE) to scroll the highlight up or down. |

| Claim Elements | Jan. 4, 2000 "CES Final" Oasis Source Code |
|---|---|
| and playing a plurality of tracks associated with the selected subcategory. | The user can scroll up and down the list of Albums. (*See* Exhs. DD and EE.) This allows for "selecting a subcategory in the second display screen."<br><br>When user presses "Play" with the "Albums" item highlighted, the CLCDMgr (*see* Exhs. FF and GG) calls CNowPlayingMgr::Play() function (see Exhs. HH and II) to clear the contents of the NowPlayingQ, and replace it with all tracks which correspond to the selected Album. This causes the selected album to begin playing. *Id.* |
| 3. The method of selecting a track as recited in claim 1<br><br>wherein the accessing at least one track comprises selecting a subcategory and | The top level of the Library Tree (*see* LibraryTree.cpp ) displays the top level categories (*i.e.*, "**Albums**," "**Artists**," "**Styles**," "**Play Lists**") on a first display screen. *See, e.g.*, Exh. BB, pp. 1-2.<br><br>When the user presses "Open" with the "Albums" category highlighted in a first display screen, the CLCDLibPage class (*see* Exhs. DD and EE) displays a current list of Albums, which is one example of "a subcategory in the second display screen."<br><br>The user can scroll up and down the list of Albums, and activate either Open, Queue, or Play buttons to select one of the albums on the list. (*See* Exhs. DD and EE.) This allows for "selecting a subcategory in the second display screen."<br><br>The user moves the selection up and down the lists by pressing the up and down arrow keys. These key-presses are handled by the CLCDMgr (see Exh. FF, p. 5) who in turn tells the CLCDLibPage (see Exhs. DD and EE) to scroll the highlight up or down. |
| adding the tracks associated with the selected subcategory to a playlist. | When the user presses the Queue button with a selected one of the albums on the list highlighted, the CLCDLibPage (*see* Exhs. DD and EE) calls the CLCDMgr::HandleQ() function (*see* Exh. , p. 6). HandleQ() in turn calls CNowPlayingMgr::Queue() and appends the items to the NowPlayingQ. (*See* Exhs. HH and II). If the NowPlayingQ were actually empty, playback would also begin immediately. *Id.* If the NowPlayingQ has actively playing content already, it continues to play. *Id.* The newly added content plays in the order it was added. *Id.* |
| 4. The method of selecting a track as recited in claim 1 wherein the accessing at least one track comprises<br><br>selecting an item in the third display screen and<br><br>playing at least one track associated with the selected item. | The top level of the Library Tree (*see* LibraryTree.cpp ) displays the top level categories (*i.e.*, "**Albums**," "**Artists**," "**Styles**," "**Play Lists**") on a first display screen. *See, e.g.*, Exh. BB, pp. 1-2.<br><br>When the user presses "Open" with the "Artists" category highlighted, the CLCDLibPage class (see Exhs. DD and EE) displays the current list of Artists on a second display screen.<br><br>The user then selects an Artist from the current list of Artists on the second display screen, and presses Open. The CLCDLibPage class (*see* Exhs. DD and EE) then populates a "third display screen" with a list of Albums for the selected Artist. Highlighting an Album and |

| Claim Elements | Jan. 4, 2000 "CES Final" Oasis Source Code |
|---|---|
| | pressing the Play button cause the CLCDMgr (*see* Exhs. FF and GG) to call the CNowPlayingMgr::Play() function (*see* Exhs. HH and II) to clear the contents of the NowPlayingQ, and replace it with all tracks which correspond to the selected Album. This causes the selected album to begin playing. *Id.* |
| 5. The method of selecting a track as recited in claim 1 wherein the accessing at least one track comprises<br><br>selecting an item in the third display screen and | The top level of the Library Tree (*see* LibraryTree.cpp ) displays the top level categories (*i.e.,* "**Albums**," "**Artists**," "**Styles**," "**Play Lists**") on a first display screen. *See, e.g.,* Exh. BB, pp. 1-2.<br><br>When the user presses "Open" with the "Artists" category highlighted, the CLCDLibPage class (see Exhs. DD and EE) displays a current list of Artists on a second display screen.<br><br>The user then selects an Artist from the current list of Artists on the second display screen, and presses Open. The CLCDLibPage class (*see* Exhs. DD and EE) then populates a "third display screen" with a list of Albums for the selected Artist. |
| adding at least one track associated with the selected item to a playlist. | Highlighting a selected Album in the current list of Albums for the selected Artist, and pressing the Queue button causes CLCDLibPage (*see* Exhs. DD and EE) to call the CLCDMgr::HandleQ() function (*see* Exhs. FF and GG). HandleQ() in turn calls CNowPlayingMgr::Queue() (*see* Exhs. HH and II) and appends the items to the NowPlayingQ. If the NowPlayingQ is actually empty, playback will also begin immediately. *Id.* If the NowPlayingQ has actively playing content already, it will continue to play. *Id.* The newly added content will play in the order it was added. *Id* |
| 6. The method of selecting a track as recited in claim 1 wherein the accessing at least one track comprises one of playing or adding to a playlist at least one track associated with a selected one of the category, subcategory, and item. | The above discussion in connection with claims 4 and/or 5 applies equally to claim 6. |
| 7. The method of selecting a track as recited in claim 1 wherein<br><br>the accessing at least one track is made after the presentation of the third display screen<br><br>by reverting back to one of the second and first display | "Reverting back" is implemented by the Close (or Back) soft key. (*See* Exh. DD, pp 1 and 3; and Exh. EE, p. 1.)<br><br>Assume a starting point of having navigated down to a third level (*e.g.,* Albums->Album1->Track3). See, for example, the discussion above in connection with claim 4, wherein: (1) a user presses "Open" with the top level "Artists" category highlighted on a "first display screen"; (2) the CLCDLibPage class (see Exhs. DD and EE) then displays a current list of Artists on a "second display screen"; (3) the user then selects an Artist from the current list of Artists on the second display screen, and presses Open, causing the CLCDLibPage class (*see* Exhs. |

| Claim Elements | Jan. 4, 2000 "CES Final" Oasis Source Code |
|---|---|
| screens,<br><br>the second display screen presented sequentially after the third display screen. | DD and EE) to populate a "third display screen" with a list of Albums for the selected Artist. The user pressing the Close soft key would then cause the CLCDLibPage class (*see* Exhs. DD and EE) to repopulate the "second display screen" with the list of Artists. Repeating pressing the Close soft will cause the CLCDLibPage class to repopulate the "first display screen" with the list of top level categories (Album, Artist, Genre, etc). At this point navigating back down the tree would follow just as in claim 4. |
| 8. The method of selecting a track as recited in claim 1 further comprising<br><br>selecting one of the items displayed in the third display screen and | The top level of the Library Tree (*see* LibraryTree.cpp ) displays the top level categories (*i.e.*, "**Albums**," "**Artists**," "**Styles**," "**Play Lists**") on a first display screen. *See, e.g.*, Exh. BB, pp. 1-2.<br><br>When the user presses "Open" with the "Artists" category highlighted, the CLCDLibPage class (see Exhs. DD and EE) displays a current list of Artists on a second display screen.<br><br>The user then selects an Artist from the current list of Artists on the second display screen, and presses Open. The CLCDLibPage class (*see* Exhs. DD and EE) then populates a "third display screen" with a list of Albums for the selected Artist. |
| presenting a listing of items associated with the selected item in a fourth sequentially presented display screen. | Now if a user presses "Open" with a selected Album item highlighted in the list of Albums, the CLCDLibPage class (*see* Exhs. DD and EE) displays the tracks for that album. This corresponds to a "fourth display screen".<br><br>Highlighting a track and pressing the Play button cause the CLCDMgr (*see* Exh. FF and G) to call CNowPlayingMgr::Play() function (*see* Exhs. HH and II) to clear the contents of the NowPlayingQ, and replace it with the track and begin playing the selected track. |
| 9. The method of selecting a track as recited in claim 1 wherein<br><br>the category genre is selected in the first display screen from available categories that include at least artist, album, and genre; and | The top level of the Library Tree (*see* LibraryTree.cpp ) displays the top level categories (*i.e.*, "**Albums**," "**Artists**," "**Styles**," "**Play Lists**") on a first display screen. *See, e.g.*, Exh. BB, pp. 1-2.<br><br>The "Styles" category is equivalent to "genre." |
| the subcategories listed in the second display screen comprise a listing of at least one genre type and one of the at least one genre type is selected. | The user selects "Styles" from the top level by highlighting it and pressing Open, which is handled by the CLCDLibPage class. *See* Exhs. DD and EE. The CLCDLibPage class then displays a list of all known Styles in a second display screen. *Id.* Style. *Id.* |

| Claim Elements | Jan. 4, 2000 "CES Final" Oasis Source Code |
|---|---|
| 10. The method of selecting a track as recited in claim 9 further comprising<br><br>displaying in the third display screen at least one album associated with the selected genre type and | See citations for claim 9, above.<br><br>The user selects "Styles" from the top level by highlighting it and pressing Open, which is handled by the CLCDLibPage class. *See* Exhs. DD and EE. The CLCDLibPage class then displays a list of all known Styles (or "genres") in a second display screen. *Id.* By highlighting a selected one of the Styles in the list and again pressing the Open soft key, the LCDLibPage class displays a list of all albums that fall within the selected Style on a "third display screen." *Id.* |
| selecting one of the at least one albums displayed in the third display screen and<br><br>presenting a listing of tracks associated with the selected album in a fourth sequentially presented display screen. | The user highlights a selected Album (*i.e.* from the list of all albums that fall within the selected Style), and pressing Open causes the CLCDLibPage class (*see* Exhs. DD and EE) to display a list of names of tracks for the selected album in a "fourth sequentially presented display screen." |
| 11. The method of selecting a track as recited in claim 1 wherein<br><br>the category artist is selected in the first display screen from available categories that include at least artist, album, and genre; | The top level of the Library Tree (*see* LibraryTree.cpp ) displays the top level categories (*i.e.*, "**Albums,**" "**Artists,**" "**Styles,**" "**Play Lists**") on a first display screen. *See, e.g.*, Exh. BB, pp. 1-2.<br><br>The "Styles" category is equivalent to "genre."<br><br>From here the user has the option of opening any of the displayed categories by pressing an "Open" softkey (*see* CLCDLib Page::Softkey1Handler() at Exh. DD, pp. 1 and 3; Exh. EE, p. 1).<br><br>The user moves the selection up and down lists by pressing up and down arrow keys, wherein the key presses are handled by the CLCDMgr (*see* Exh. FF, p. 5) which in turn tells the CLCDLibPage (*see* Exhs. DD and EE) to scroll the highlight up or down.<br><br>When a user presses "Open" with the top level "Artists" category highlighted on a "first display screen" (that includes the above top-level categories), the CLCDLibPage class (*see* Exhs. DD and EE) displays a current list of Artists on a "second display screen." |
| the subcategories listed in the second display screen comprise a listing of names of artists and a first artist name is selected; and. | When a user presses "Open" with the top level "Artists" category highlighted on a "first display screen" (that includes the above top-level categories), the CLCDLibPage class (*see* Exhs. DD and EE) displays a current list of Artists on a "second display screen."<br><br>The user then selects an Artist from the current list of Artists on the second display screen, and presses Open, causing the CLCDLibPage class (*see* Exhs. DD and EE) to populate a "third display screen" with a list of Albums for the selected Artist. |
| the items displayed in the | The user then selects an Artist from the current list of Artists on the |

| Claim Elements | Jan. 4, 2000 "CES Final" Oasis Source Code |
|---|---|
| third display screen comprises at least one album associated with the first artist name | second display screen, and presses Open, causing the CLCDLibPage class (*see* Exhs. DD and EE) to populate a "third display screen" with a list of Albums for the selected Artist. |
| 12. The method of selecting a track as recited in claim 1 wherein<br><br>the track is a music track, | A plurality of musical "tracks" are stored in a computer-readable medium accessed via the library tree. *See* Exhs. B and C. For example, the file LibraryTree.cpp refers to the "tracks directory. *See* Exh. BB, p. 2. |
| accessing at least one track comprises accessing a track title in the third display screen, and the track is played in response to the access. | The top level of the Library Tree (*see* LibraryTree.cpp ) displays the top level categories (*i.e.*, "**Albums**," "**Artists**," "**Styles**," "**Play Lists**") on a first display screen. *See, e.g.*, Exh. BB, pp. 1-2.<br><br>Pressing Open with the "Albums" item highlighted, the CLCDLibPage class (*see* Exhs. DD and EE) displays the current list of Albums. This would correspond to a second display screen.<br><br>When the user again presses "Open" with a particular album highlighted, the CLCDLibPage class (*see* Exhs. DD and EE) then displays a list of tracks corresponding to the selected album on a "third display screen."<br><br>The user then highlights a selected tracks from the list, and pressing the Play Button causes the CLCDMgr (*see* Exhs. FF and GG) to call the CNowPlayingMgr::Play() function (*see* Exhs. HH and II) to clear the contents of the NowPlayingQ, and replace it with the track that is currently selected. At this point the track begins playing. *Id.* |
| 13. The method of selecting a track as recited in claim 1 wherein<br><br>receipt of the selection in the first display screen results in an automatic transition of the first display screen into the second display screen and<br><br>receipt of the selection in the second display screen results in an automatic transition of the second display screen into the third display screen. | The top level of the Library Tree (*see* LibraryTree.cpp ) displays the top level categories (*i.e.*, "**Albums**," "**Artists**," "**Styles**," "**Play Lists**") on a first display screen. *See, e.g.*, Exh. BB, pp. 1-2.<br><br>Highlighting any item and pressing "Open" causes CLCDLibPage to automatically display elements in a second level for the selected category. Highlighting any item and pressing "Open" again causes CLCDLibPage to automatically display the third level. |
| 14. The method of selecting a track as recited in claim 1 | The top level of the Library Tree (*see* LibraryTree.cpp ) displays the top level categories (*i.e.*, "**Albums**," "**Artists**," "**Styles**," "**Play Lists**") |

| Claim Elements | Jan. 4, 2000 "CES Final" Oasis Source Code |
|---|---|
| wherein the category selected in the first display screen is from a top level of the hierarchy. | on a first display screen. *See, e.g.,* Exh. BB, pp. 1-2.<br>The user has the option of opening any of the displayed top level categories by pressing the "Open" softkey. (*See* CLCDLibPage::Softkey1Handler() at Exhs. DD and EE). |
| 15. The method of selecting a track as recited in claim 1 wherein<br>the category selected in the first display screen is a category from a level at least one level below the top level of the hierarchy. | To the extent that the "first display screen" is not limited to a display screen presented initially upon start-up of the system, the December 14, 1999 Nomad Jukebox source code implemented this claim.<br>The top level of the Library Tree (*see* LibraryTree.cpp ) displays the top level categories (*i.e.,* "**Albums**," "**Artists**," "**Styles**," "**Play Lists**") on a first display screen. *See, e.g.,* Exh. BB, pp. 1-2.<br>The has the option of opening any of the displayed top level categories by pressing the "Open" softkey. (*See* CLCDLibPage::Softkey1Handler() at Exhs. DD and EE).<br>If the user presses "Open" with the "Albums" item highlighted, the CLCDLibPage class (*see* Exhs. DD and EE) displays the current list of Albums on another "display screen." If this "display screen" showing the current list of Albums may be considered to be a "first display screen," even though it is not the first screen upon start-up of the system, then "the category selected in the first display screen is a category from a level at least one level below the top level of the hierarchy." |
| 16. The method of selecting a track as recited in claim 1 wherein<br>the plurality of categories comprise a list of artist names, | The top level of the Library Tree (*see* LibraryTree.cpp ) displays the top level categories (*i.e.,* "**Albums**," "**Artists**," "**Styles**," "**Play Lists**") on a first display screen. *See, e.g.,* Exh. BB, pp. 1-2.<br>When the user presses "Open" with the "Artists" category highlighted, the CLCDLibPage class (see Exhs. DD and EE) displays a current list of Artist names on a second display screen. |
| the plurality of subcategories comprise a list of album names and | The user then selects an Artist name from the current list of Artists on the second display screen, and presses Open. The CLCDLibPage class (*see* Exhs. DD and EE) then populates a display screen with a list of Album names for the selected Artist. |
| the plurality of items comprise a list of track names. | Now if a user presses "Open" with a selected Album item highlighted in the list of Albums, the CLCDLibPage class (*see* Exhs. DD and EE) displays the tracks for that album.<br>Highlighting a track and pressing the Play button cause the CLCDMgr (*see* Exh. FF and G) to call CNowPlayingMgr::Play() function (*see* Exhs. HH and II) to clear the contents of the NowPlayingQ, and replace it with the track and begin playing the selected track. |
| 17. The method of selecting a track as recited in claim 1 wherein the | The source code files LibraryTree.cpp and LibraryTree.h implement a hierarchy in the form of a library tree. *See* Exhs. B and C. The library tree stores human-readable information that is shown in a user |

| Claim Elements | Jan. 4, 2000 "CES Final" Oasis Source Code |
|---|---|
| hierarchy is an overlapping hierarchy having a plurality of categories that include items, and wherein at least one of the items is included in more than one of the categories. | interface, and also stores the actual details of where to find a particular track in the file system. *Id.* A display screen is displayed for every level of the hierarchy. *Id.* Thus, the entire display is replicated by the library tree. *Id.* This way the user-interface code traverses the tree and displays any items(s) at the current node. *Id.*<br>The source code files LibraryTree.cpp and LibraryTree.h indicate that the library tree is an overlapping hierarchy. *See* Exhs. B and C. |
| 18. The method of selecting a track as recited in claim 17 wherein the items comprise a plurality of track names, wherein at least one of the track names is included in more than one of the categories, whereby the least one track name may be accessed in at least two different ways by starting with different ones of the categories. | The source code files LibraryTree.cpp and LibraryTree.h implement a hierarchy in the form of a library tree. *See* Exhs. B and C. The library tree stores human-readable information that is shown in a user interface, and also stores the actual details of where to find a particular track in the file system. *Id.* A display screen is displayed for every level of the hierarchy. *Id.* Thus, the entire display is replicated by the library tree. *Id.* This way the user-interface code traverses the tree and displays any items(s) at the current node. *Id.*<br>The source code files LibraryTree.cpp and LibraryTree.h indicate that the library tree is an overlapping hierarchy. *See* Exhs. B and C.<br><br>The source code files LibraryTree.cpp and LibraryTree.h indicate that tracks may be accessed in different ways. *See* Exhs. B and C. |
| 19. The method of selecting a track as recited in claim 1 wherein the hierarchy comprises an implementation of a tree-structure. | The source code files LibraryTree.cpp and LibraryTree.h implement a hierarchy in the form of a library tree. *See* Exhs. B and C. The library tree stores human-readable information that is shown in a user interface, and also stores the actual details of where to find a particular track in the file system. *Id.* A display screen is displayed for every level of the hierarchy. *Id.* Thus, the entire display is replicated by the library tree. *Id.* This way the user-interface code traverses the tree and displays any items(s) at the current node. *Id.*<br>A plurality of musical "tracks" are stored in a computer-readable medium accessed via the library tree. *See* file LibraryTree.cpp at Exh. BB, p. 2 (referring to a "tracks directory"); *see also* Exh. CC.<br>As indicated by developer notes in the source code, "Class CLibTreeMgr creates and makes modifications as necessary to a tree structure whose purpose is to make the track data on disk logically and quickly navigable. It is currently rebuilt every time the system boots." *See* Exh. BB, p. 1; Exh. CC, p. 1. |
| 20. The method of selecting a track as recited in claim 19 wherein the tree-structure is organized based on metadata associated with the tracks. | The source code files LibraryTree.cpp and LibraryTree.h implement a hierarchy in the form of a library tree. *See* Exhs. B and C. The library tree stores human-readable information that is shown in a user interface, and also stores the actual details of where to find a particular track in the file system. *Id.* A display screen is displayed for every level of the hierarchy. *Id.* Thus, the entire display is replicated by the library tree. *Id.* This way the user-interface code traverses the tree and displays any items(s) at the current node. *Id.* |

| Claim Elements | Jan. 4, 2000 "CES Final" Oasis Source Code |
|---|---|
| | A plurality of musical "tracks" are stored in a computer-readable medium accessed via the library tree. *See* file LibraryTree.cpp at Exh. BB, p. 2 (referring to a "tracks directory"); *see also* Exh. CC.<br>As indicated by developer notes in the source code, "Class CLibTreeMgr creates and makes modifications as necessary to a tree structure whose purpose is to make the track data on disk logically and quickly navigable. It is currently rebuilt every time the system boots." *See* Exh. BB, p. 1; Exh. CC, p. 1. |
| 21. The method of selecting a track as recited in claim 3 wherein the playlist is an active queue list of songs that is currently being played. | When the user presses the Queue button with a selected one of the tracks on the list highlighted, the CLCDLibPage (*see* Exhs. DD and EE) calls the CLCDMgr::HandleQ() function (*see* Exh. FF, p. 7). HandleQ() in turn calls CNowPlayingMgr::Queue() and appends the items to the NowPlayingQ. (*See* Exhs. HH, II). If the NowPlayingQ were actually empty, playback would also begin immediately. *Id.* If the NowPlayingQ has actively playing content already, it continues to play. *Id.* The newly added content plays in the order it was added. *Id.* |
| 22. The method of selecting a track as recited in claim 5 wherein the playlist is an active queue list of songs that is currently being played. | When the user presses the Queue button with a selected one of the items on the list highlighted, the CLCDLibPage (*see* Exhs. DD and EE) calls the CLCDMgr::HandleQ() function (*see* Exh. FF, p. 7). HandleQ() in turn calls CNowPlayingMgr::Queue() and appends the items to the NowPlayingQ. (*See* Exhs. HH, II). If the NowPlayingQ were actually empty, playback would also begin immediately. *Id.* If the NowPlayingQ has actively playing content already, it continues to play. *Id.* The newly added content plays in the order it was added. *Id.* |
| 23. The method of selecting a track as recited in claim 5<br>wherein the selected item in the third display screen is associated with a plurality of tracks, and | The top level of the Library Tree (*see* LibraryTree.cpp ) displays the top level categories (*i.e.*, "**Albums**," "**Artists**," "**Styles**," "**Play Lists**") on a first display screen. *See, e.g.,* Exh. BB, pp. 1-2.<br>The "Styles" category is equivalent to "genre."<br>The user selects "Styles" from the top level by highlighting it and pressing Open, which is handled by the CLCDLibPage class. *See* Exhs. DD and EE. The CLCDLibPage class then displays a list of all known Styles in a second display screen. *Id.*<br>By highlighting a selected one of the Styles in the list and pressing Open, the LCDLibPage class displays a list of all album namess that fall within the selected Style on a third display screen. Id. |
| wherein the plurality of tracks associated with the selected item are added to the playlist. | When the user presses Queue with a selected one of the albums highlighted, the CLCDLibPage (*see* Exhs. DD and EE) calls the CLCDMgr::HandleQ() function (*see* Exh. FF, p. 7). HandleQ() in turn calls CNowPlayingMgr::Queue() and appends the tracks associated with the selected album to the NowPlayingQ. (*See* Exhs. HH, II). |
| 24. The method of selecting a track as recited | When the user presses the Queue button with a selected one of the albums on the list highlighted, the CLCDLibPage (*see* Exhs. DD and |

| Claim Elements | Jan. 4, 2000 "CES Final" Oasis Source Code |
|---|---|
| in claim 23 wherein the playlist is an active queue list of songs that is currently being played. | EE) calls the CLCDMgr::HandleQ() function (*see* Exh. FF, p. 7). HandleQ() in turn calls CNowPlayingMgr::Queue() and appends the items to the NowPlayingQ. (*See* Exhs. HH, II). If the NowPlayingQ were actually empty, playback would also begin immediately. *Id.* If the NowPlayingQ has actively playing content already, it continues to play. *Id.* The newly added content plays in the order it was added. *Id.* |
| 25. The method of selecting a track as recited in claim 5 wherein the selected item in the third display screen is a selected album name, | The top level of the Library Tree (*see* LibraryTree.cpp ) displays the top level categories (*i.e.*, "**Albums**," "**Artists**," "**Styles**," "**Play Lists**") on a first display screen. *See, e.g.*, Exh. BB, pp. 1-2.<br>The "Styles" category is equivalent to "genre."<br>The user selects "Styles" from the top level by highlighting it and pressing Open, which is handled by the CLCDLibPage class. *See* Exhs. DD and EE. The CLCDLibPage class then displays a list of all known Styles in a second display screen. *Id.*<br>By highlighting a selected one of the Styles in the list and again pressing Open, the LCDLibPage class displays a list of all album names that fall within the selected Style on a third display screen. Id. |
| and wherein the accessing at least one track comprises adding a plurality of tracks associated with the selected album name to a playlist. | When the user presses Queue with a selected one of the albums names highlighted, the CLCDLibPage (*see* Exhs. DD and EE) calls the CLCDMgr::HandleQ() function (*see* Exh. FF, p. 7). HandleQ() in turn calls CNowPlayingMgr::Queue() and appends the tracks associated with the selected album name to the NowPlayingQ. (*see* Exhs. HH, II). |
| 26. The method of selecting a track as recited in claim 25 wherein the playlist is an active queue list of songs that is currently being played. | When the user presses Queue with a selected one of the albums highlighted, the CLCDLibPage (*see* Exhs. DD and EE) calls the CLCDMgr::HandleQ() function (*see* Exh. FF, p. 7). HandleQ() in turn calls CNowPlayingMgr::Queue() and appends the tracks associated with the selected album name to the NowPlayingQ. (*See* Exhs. HH, II). If the NowPlayingQ were actually empty, playback would also begin immediately. *Id.* If the NowPlayingQ has actively playing content already, it continues to play. *Id.* The newly added content plays in the order it was added. *Id.* |
| 27. The method of selecting a track as recited in claim 1 wherein:<br>the category album is selected in the first display screen from available categories that include at least artist and album; | The top level of the Library Tree (*see* LibraryTree.cpp ) displays the top level categories (*i.e.*, "**Albums**," "**Artists**," "**Styles**," "**Play Lists**") on a first display screen. *See, e.g.*, Exh. BB, pp. 1-2.<br>When the user presses Open with the "Albums" category highlighted, the CLCDLibPage class (*see* Exhs. DD and EE) displays a current list of Album names |
| the subcategories listed in the second display screen comprise a listing of album | When the user presses Open with the "Albums" category highlighted, the CLCDLibPage class (*see* Exhs. DD and EE) displays a current list of Album names |

| Claim Elements | Jan. 4, 2000 "CES Final" Oasis Source Code |
|---|---|
| names and one of the album names is selected; and | The user can scroll up and down the list of Album names, and activate either Open, Queue, or Play buttons to select one of the albums on the list. (*See* Exhs. DD and EE.) |
| the accessing at least one track comprises playing a plurality of tracks associated with the selected album name. | When user presses Play with a selected album name highlighted, the CLCDMgr (*see* Exhs. FF and GG) calls CNowPlayingMgr::Play() function (see Exhs. HH and II) to clear the contents of the NowPlayingQ, and replace it with all tracks which correspond to the selected Album. This causes the selected album to begin playing. *Id.* |
| 28. The method of selecting a track as recited in claim 1 wherein:<br>the category album is selected in the first display screen from available categories that include at least artist and album; | The top level of the Library Tree (*see* LibraryTree.cpp ) displays the top level categories (*i.e.*, "**Albums**," "**Artists**," "**Styles**," "**Play Lists**") on a first display screen. *See, e.g.*, Exh. BB, pp. 1-2.<br>When the user presses Open with the "Albums" category highlighted, the CLCDLibPage class (*see* Exhs. DD and EE) displays a current list of Album names on a second display screen. |
| the subcategories listed in the second display screen comprise a listing of album names and one of the album names is selected; and | When the user presses Open with the Albums category highlighted, the CLCDLibPage class (*see* Exhs. DD and EE) displays a current list of Album names on a second display screen.<br>The user can scroll up and down the list of Album names on the second display screen, and activate either Open, Queue, or Play buttons to select one of the albums on the list. (*See* Exhs. DD and EE.) |
| the accessing at least one track comprises adding a plurality of tracks associated with the selected album name to a playlist. | When the user presses Queue with a selected one of the album names highlighted on the list, the CLCDLibPage (*see* Exhs. DD and EE) calls the CLCDMgr::HandleQ() function (*see* Exh. FF, p. 7). HandleQ() in turn calls CNowPlayingMgr::Queue() and appends the tracks associated with the selected album name to the NowPlayingQ. (*See* Exhs. HH, II). |
| 29. The method of selecting a track as recited in claim 28 wherein the playlist is an active queue list of songs that is currently being played. | When the user presses Queue with a selected one of the album names highlighted, the CLCDLibPage (*see* Exhs. DD and EE) calls the CLCDMgr::HandleQ() function (*see* Exh. FF, p. 7). HandleQ() in turn calls CNowPlayingMgr::Queue() and appends the tracks associated with the selected album name to the NowPlayingQ. (*See* Exhs. HH, II). If the NowPlayingQ were actually empty, playback would also begin immediately. *Id.* If the NowPlayingQ has actively playing content already, it continues to play. *Id.* The newly added content plays in the order it was added. *Id.* |

| Claim Elements | Jan. 4, 2000 "CES Final" Oasis Source Code |
|---|---|
| 30. (New) The method of selecting a track as recited in claim 1 wherein:<br>the category genre is selected in the first display screen from available categories that include at least artist, album, and genre; | The top level of the Library Tree (*see* LibraryTree.cpp ) displays the top level categories (*i.e.*, "**Albums**," "**Artists**," "**Styles**," "**Play Lists**") on a first display screen. *See, e.g.*, Exh. BB, pp. 1-2.<br>The "Styles" category is equivalent to "genre." |
| the subcategories listed in the second display screen comprise a listing of a plurality of genre types, and one of one genre types is selected; | The user selects Styles from the first display screen by highlighting it and pressing Open, which is handled by the CLCDLibPage class. *See* Exhs. DD and EE. The CLCDLibPage class then displays a list of all known Styles in a second display screen. *Id.* |
| the items displayed in the third display screen comprise a listing of a plurality of album names associated with the selected genre type, and one of the album names is selected; | By highlighting a selected one of the Styles in the list on the second display screen and again pressing Open, the LCDLibPage class displays a list of all album names that fall within the selected Style on a third display screen. *Id.* |
| the accessing at least one track comprises adding a plurality of tracks associated with the selected album name to a playlist. | When the user presses Queue with a selected one of the album names highlighted, the CLCDLibPage (*see* Exhs. DD and EE) calls the CLCDMgr::HandleQ() function (*see* Exh. FF, p. 7). HandleQ() in turn calls CNowPlayingMgr::Queue() and appends the tracks associated with the selected album name to the NowPlayingQ. (*see* Exhs. HH, II). |
| 31. The method of selecting a track as recited in claim 30 wherein the playlist is an active queue list of songs that is currently being played. | When the user presses Queue with a selected one of the album names highlighted, the CLCDLibPage (*see* Exhs. DD and EE) calls the CLCDMgr::HandleQ() function (*see* Exh. FF, p. 7). HandleQ() in turn calls CNowPlayingMgr::Queue() and appends the tracks associated with the selected album name to the NowPlayingQ. (*See* Exhs. HH, II). If the NowPlayingQ were actually empty, playback would also begin immediately. *Id.* If the NowPlayingQ has actively playing content already, it continues to play. *Id.* The newly added content plays in the order it was added. *Id.* |

| Claim Elements | Jan. 4, 2000 "CES Final" Oasis Source Code |
|---|---|
| 32. The method of selecting a track as recited in claim 1 wherein:<br>the category artist is selected in the first display screen from available categories that include at least artist, album, and genre; | The top level of the Library Tree (*see* LibraryTree.cpp ) displays the top level categories (*i.e.*, "**Albums**," "**Artists**," "**Styles**," "**Play Lists**") on a first display screen. *See, e.g.*, Exh. BB, pp. 1-2.<br>From here the user has the option of opening any of the displayed categories by pressing Open (*see* CLCDLib Page::Softkey1Handler() at Exh. D, pp. 1 and 3; Exh. E, p. 1).<br>When a user presses "Open" with the top level "Artists" category highlighted (that includes the above top-level categories), the CLCDLibPage class (see Exhs. DD and EE) displays a current list of Artists on a second display screen. |
| the subcategories listed in the second display screen comprise a listing of artist names, and one of the listed artist names is selected; | When a user presses "Open" with the top level "Artists" category highlighted (that includes the above top-level categories), the CLCDLibPage class (see Exhs. DD and EE) displays a current list of Artists on a second display screen. |
| the items displayed in the third display screen comprise a listing of album names associated with the selected artist name, and one of the listed album names is selected; and | The user then selects an Artist name from the Artist names on the second display screen, and presses Open, causing the CLCDLibPage class (see Exhs. DD and EE) to populate a third display screen with a list of Album names for the selected Artist. |
| the accessing at least one track comprises adding a plurality of tracks associated with the selected album name to a playlist. | When the user presses Queue with a selected one of the album names highlighted, the CLCDLibPage (*see* Exhs. DD and EE) calls the CLCDMgr::HandleQ() function (*see* Exh. FF, p. 7). HandleQ() in turn calls CNowPlayingMgr::Queue() and appends the tracks associated with the selected album name to the NowPlayingQ. (*see* Exhs. HH, II). |
| 33. The method of selecting a track as recited in claim 32 wherein the playlist is an active queue list of songs that is currently being played. | When the user presses Queue with a selected one of the album names on the list highlighted, the CLCDLibPage (*see* Exhs. DD and EE) calls the CLCDMgr::HandleQ() function (*see* Exh. FF, p. 7). HandleQ() in turn calls CNowPlayingMgr::Queue() and appends the tracks associated with the selected album name to the NowPlayingQ. (*See* Exhs. HH, II). If the NowPlayingQ were actually empty, playback would also begin immediately. *Id.* If the NowPlayingQ has actively playing content already, it continues to play. *Id.* The newly added content plays in the order it was added. *Id.* |

60. Table B, above, demonstrates that the Jan. 4, 2000 Oasis source code (*see* Exhs. AA through OO) - which was embedded as fully operative firmware in the NOMAD® Jukebox portable media players presented at CES 2000 - implemented each and every element of claims 1-33.

61. The NOMAD® Jukebox portable media players presented at CES 2000 were thoroughly tested at least as early as Jan. 4, 2000. As explained above, the bootloader problem - which I had previously discussed at the Oasis Engineering Meeting on Dec. 9, 1999 – was solved at least as early as Dec. 28, 1999. *See* Exh. X; *see also* Exh. W, pp. 15-16; *see also* Exh. Z, p. 75 (showing a check-in by Howard Egan on 12/23/99 with notes indicating "Latest boot loader modifications and inclusion of some new hard coded music.") As I explained above, the bootloader was not essential to demonstrating that the Oasis operating system (including all of the functionality claims 1-33 of the `433 Patent) was working for its intended purpose. Nevertheless, the NOMAD® Jukebox portable media players presented at CES 2000 were able to power up and function without JTAG control, and completely free of any test set-up assistance. Moreover, the Oasis operating system in the NOMAD® Jukebox portable media players presented at CES 2000 implemented all of the functionality of claims 1-33 of the `433 Patent, and it worked for its intended purpose, which was to provide a user of a portable media player with an intuitive and efficient interface for accessing and playing songs.

62. The Nomad Jukebox "Public Demonstration" document clearly shows that the NOMAD® Jukeboxes demonstrated at CES were working for their intended purpose because it reflects a very high level of confidence in being ready to publicly demonstrate the features covered by claims 1-33 of the `433 Patent, which are manifest in the below description:

- "Lets start with the LibraryKey – show that this skips between the **top library** Screen and the **'now playing' Screen**"
- "SoftKeys – repeat that the **softkeys are labeled depending on the screen**, and for example, when searching the library, they are **used to expand the view via different categories**";
- "Show the use of the ScrollKeys **by moving up and down lists of albums, styles, artists or tracks**";
- "Touch the LibraryKey again to return to the **top Library screen** and **select an album**. (let the audience **choose a style**). Show how to **play this directly from the Library by pressing** the **PLAY** transport key"
- "You find and **select** what you want to hear and simply **press Play**'"
- "Remember: if **the PlayKey is pressed** when the Library Screen is active, it will **immediately play the selection** and will not act as pause"
- "**Building and playing playlists** is an important part of the JukeBox's function. We've shown how easy it is to **immediately Play any Album from the library by selecting and using the PlayKey**."
- "**To build up a list of selections**, instead of pressing the PlayKey when searching through the library, **press the QueueKey** ... This **will add your selection to the currently playing list**, and you can check this by looking at the list view of the PlayScreens. After any listening session, **you can choose to save all the music you have been listening to as a playlist** for future occasions."

*See* Exh. Y, p. 3 (emphasis added).

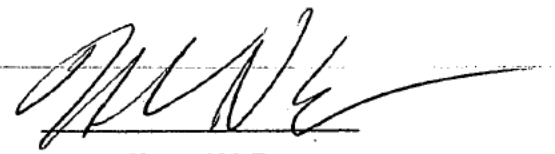
63. Given that the above "Public Demonstration" instructions were written for the purpose of demonstrating a functional NOMAD® Jukebox having an Oasis operating system compiled as firmware from the Jan. 4, 2000 Oasis source code (which implements claims 1-33 of the `433 Patent), it is clear that the NOMAD® Jukebox portable media players demonstrated at CES 2000 were working for the intended purpose of the inventions recited in claims 1-33 of the `433 Patent. For all of the reasons explained above, and based on all of the evidence attached to this declaration, I submit that the NOMAD® Jukebox portable media players demonstrated at CES 2000 were fully reduced to practice *at least as early as* Jan. 4, 2000, and they implemented all of the elements of claims 1-33.

64. For all of the reasons explained above, and based on all of the evidence attached to this declaration, the inventors of the `433 Patent had conceived of the inventions recited in claims 1-33 of the `433 Patent at least as early as Dec. 14, 1999, and the Oasis R&D team (including myself) was diligent in its effort to reduce to practice the inventions of claims 1-33 throughout the entire time period starting from Dec. 14, 1999 spanning through Jan. 4, 2000.

65. I note that additional evidence in the form of email, design documents, and schematics may exist in archives from the computers of Lee Morse (Project Manager), Dan Freeman, Andrei Veltchev, and Dave Bristow. However, time did not permit me to retrieve this from company tape archives. Therefore, I reserve the right to supplement this declaration if additional non-cumulative evidence becomes available to me at a later date.

I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and that these statements were made with knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code.

Executed May 29, 2010 at Capitola, California.



Howard N. Egan.

# EXHIBIT
# A

```
 All files in Oasis project at December 14, 1999.


 Volume in drive C has no label.
 Volume Serial Number is 9291-D72E

 Directory of C:\Projects\OASIS_Review\oasis\BufferPools

05/03/2010  02:24 PM             6,611 BufferPools.cpp

 Directory of C:\Projects\OASIS_Review\oasis\BufferPools

10/20/1999  06:29 AM             3,992 BufferPools.h
               2 File(s)         10,603 bytes

 Directory of C:\Projects\OASIS_Review\oasis\cl7211

05/03/2010  02:24 PM            21,813 cl7211.h
09/09/1999  06:13 PM             7,591 machine.h

 Directory of C:\Projects\OASIS_Review\oasis\cl7211

05/03/2010  02:24 PM             3,136 interrupt.c
               3 File(s)         32,540 bytes

 Directory of C:\Projects\OASIS_Review\oasis\DSPManager

05/03/2010  02:24 PM            12,124 CDspIO.cpp
05/03/2010  02:24 PM            11,413 CDspManager.cpp

 Directory of C:\Projects\OASIS_Review\oasis\DSPManager

05/03/2010  02:24 PM             3,073 CDspIO.h
05/03/2010  02:24 PM             5,141 CDspManager.h
05/03/2010  02:24 PM            21,255 dspboot.h
05/03/2010  02:24 PM            29,380 dspcode.h
               6 File(s)         82,386 bytes

 Directory of C:\Projects\OASIS_Review\oasis\FileSystem

05/03/2010  02:24 PM             5,948 CAttributes.cpp
05/03/2010  02:24 PM             8,213 CDirInode.cpp
05/03/2010  02:24 PM            26,863 CINode.cpp
05/03/2010  02:24 PM            11,991 CPartitionIo.cpp
05/03/2010  02:24 PM            15,285 CStorageMap.cpp
05/03/2010  02:24 PM             8,432 CSuperBlock.cpp
05/03/2010  02:24 PM             6,731 FileSystem.cpp
05/03/2010  02:24 PM             3,378 HardFileData.cpp
05/03/2010  02:24 PM             7,815 TFsTestForm.cpp
05/03/2010  02:24 PM             4,296 TransferAgent.cpp

 Directory of C:\Projects\OASIS_Review\oasis\FileSystem

05/03/2010  02:24 PM             2,113 CAttributes.h
05/03/2010  02:24 PM             1,340 CDirInode.h
05/03/2010  02:24 PM             5,923 CINode.h
05/03/2010  02:24 PM             3,511 CPartitionIo.h
05/03/2010  02:24 PM             2,380 CStorageMap.h
05/03/2010  02:24 PM             2,320 CSuperBlock.h
05/03/2010  02:24 PM             2,178 FileSystem.h
11/11/1999  02:51 PM               230 FsInternals.h
05/03/2010  02:24 PM             1,454 FsTypes.h
11/11/1999  02:51 PM         3,034,058 funky.h
11/11/1999  02:51 PM               246 HardFileData.h
05/03/2010  02:24 PM             1,455 TransferAgent.h
              22 File(s)      3,156,160 bytes
```

```
 Directory of C:\Projects\OASIS_Review\oasis\flash

09/21/1998  06:42 AM             1,410 amdflash.h
05/03/2010  02:24 PM             7,982 flash.h
11/01/1998  06:27 PM             2,532 intelflash.h
12/08/1998  10:58 AM             1,353 intelgchipflash.h

 Directory of C:\Projects\OASIS_Review\oasis\flash

12/29/1998  05:57 PM            29,552 amdflash.c
05/03/2010  02:24 PM            37,612 flash.c
05/03/2010  02:24 PM            35,498 intelflash.c
05/03/2010  02:24 PM            15,897 intelgchipflash.c
               8 File(s)        131,836 bytes

 Directory of C:\Projects\OASIS_Review\oasis\FrontPanelUI

05/03/2010  02:24 PM            33,694 CStr.cpp
05/03/2010  02:24 PM               438 LCDFunctionsPage.cpp
05/03/2010  02:24 PM             1,245 LCDGamesPage.cpp
05/03/2010  02:24 PM             4,499 LCDLibPage.cpp
05/03/2010  02:24 PM             7,809 LCDListBox.cpp
05/03/2010  02:24 PM             6,859 LCDListeningPage.cpp
05/03/2010  02:24 PM             1,025 LCDMenu.cpp
05/03/2010  02:24 PM             1,596 LCDMenuPage.cpp
05/03/2010  02:24 PM             7,250 LCDMgr.cpp
05/03/2010  02:24 PM             3,468 LCDPage.cpp
05/03/2010  02:24 PM             3,335 LCDQListPage.cpp
05/03/2010  02:24 PM               993 LCDSplashPage.cpp
05/03/2010  02:24 PM             1,228 LCDTrackInfoPage.cpp
05/03/2010  02:24 PM               928 LCDTracksPage.cpp

 Directory of C:\Projects\OASIS_Review\oasis\FrontPanelUI

05/03/2010  02:24 PM            13,551 CStr.h
11/15/1999  01:50 PM             2,051 GeneralTypes.h
05/03/2010  02:24 PM             2,108 GrCustom.h
05/03/2010  02:24 PM               892 GrResources.h
10/06/1999  06:23 AM               242 LCDFunctionsPage.h
11/15/1999  01:50 PM               702 LCDGamesPage.h
05/03/2010  02:24 PM               719 LCDLibPage.h
05/03/2010  02:24 PM             2,345 LCDListBox.h
05/03/2010  02:24 PM               355 LCDListeningPage.h
10/06/1999  06:24 AM             1,052 LCDMenu.h
05/03/2010  02:24 PM               291 LCDMenuPage.h
05/03/2010  02:24 PM             2,723 LCDMgr.h
05/03/2010  02:24 PM             2,766 LCDPage.h
05/03/2010  02:24 PM               409 LCDQListPage.h
11/15/1999  01:50 PM               351 LCDSplashPage.h
05/03/2010  02:24 PM               497 LCDTrackInfoPage.h
10/06/1999  06:26 AM               475 LCDTracksPage.h
11/15/1999  01:50 PM               152 StdInclude.h

 Directory of C:\Projects\OASIS_Review\oasis\FrontPanelUI

05/03/2010  02:24 PM             8,153 GrCustom.c
05/03/2010  02:24 PM           124,708 GrResources.c
              34 File(s)       238,909 bytes

 Directory of C:\Projects\OASIS_Review\oasis\graphics

05/03/2010  02:24 PM            14,111 Graphics.h
01/14/1994  02:59 PM               938 GrCustom.h
10/06/1999  06:49 AM               510 GrDriver.h
10/06/1999  06:57 AM             3,019 GrEngine.h
```

```
08/05/1998  12:01 PM             7,940 GrLcdConfig.h
01/14/1994  02:59 PM               366 GrResources.h

 Directory of C:\Projects\OASIS_Review\oasis\graphics

05/03/2010  02:24 PM            38,999 Graphics.c
01/14/1994  02:59 PM             7,645 GrCustom.c
05/03/2010  02:24 PM            50,734 GrEngine.c
10/06/1999  02:42 PM            26,068 GrLcdDriver.c
01/14/1994  02:59 PM            45,936 GrResources.c
              11 File(s)        196,266 bytes

 Directory of C:\Projects\OASIS_Review\oasis\hw

08/05/1999  02:27 PM             1,657 crc.h
05/03/2010  02:24 PM             1,364 MemTest.h

 Directory of C:\Projects\OASIS_Review\oasis\hw

08/05/1999  02:27 PM             9,779 crc.c
05/03/2010  02:24 PM             3,523 MemTest.c
               4 File(s)         16,323 bytes

 Directory of C:\Projects\OASIS_Review\oasis\IDE

05/03/2010  02:24 PM             3,298 blockdrv.h
05/03/2010  02:24 PM             2,518 COasisSmartIDE.h
05/03/2010  02:24 PM            10,551 idedrv.h
               3 File(s)         16,367 bytes

 Directory of C:\Projects\OASIS_Review\oasis\Include

11/11/1999  04:15 PM             4,934 mpgaudio.h
11/11/1999  04:16 PM               627 mpgdata.h
               2 File(s)          5,561 bytes

 Directory of C:\Projects\OASIS_Review\oasis\kernel

10/01/1999  01:17 PM               752 CTask.h
08/05/1999  11:45 AM             4,781 Exception.h
05/03/2010  02:24 PM             3,866 Kernel.h
05/03/2010  02:24 PM             7,545 KernelP.h
05/03/2010  02:24 PM             2,524 KernelTypes.h

 Directory of C:\Projects\OASIS_Review\oasis\kernel

09/03/1998  07:52 PM             1,366 Exception.c
05/03/2010  02:24 PM            51,921 Kernel.c
               7 File(s)         72,755 bytes

 Directory of C:\Projects\OASIS_Review\oasis\LibMgr

05/03/2010  02:24 PM             3,346 LibElement.cpp
05/03/2010  02:24 PM            12,185 LibraryTree.cpp
05/03/2010  02:24 PM             8,340 NowPlayingQ.cpp

 Directory of C:\Projects\OASIS_Review\oasis\LibMgr

05/03/2010  02:24 PM             2,512 LibElement.h
05/03/2010  02:24 PM             2,011 LibraryTree.h
05/03/2010  02:24 PM             2,417 NowPlayingQ.h
05/03/2010  02:24 PM             2,049 TestTrkAttrText.h
               7 File(s)         32,860 bytes

 Directory of C:\Projects\OASIS_Review\oasis\main
```

```
09/28/1999  07:31 AM               790 BufferTypes.h
12/13/1999  04:33 PM             7,724 cl7211Config.h
09/09/1999  05:48 PM             2,790 flashconfig.h
09/09/1999  06:40 PM             1,019 hwconfig.h
12/14/1999  08:36 AM             2,748 KernelConfig.h
05/03/2010  02:24 PM               436 MessageTypes.h
12/13/1999  12:54 PM             3,021 OasisHW.h
05/03/2010  02:24 PM               547 PortIdents.h

 Directory of C:\Projects\OASIS_Review\oasis\main

12/13/1999  04:31 PM             5,511 OasisHWInit.c
               9 File(s)         24,586 bytes

 Directory of C:\Projects\OASIS_Review\oasis\oasishw

05/03/2010  02:24 PM             2,147 ButtonScan.h
05/03/2010  02:24 PM             4,184 KS0713LCD.h

 Directory of C:\Projects\OASIS_Review\oasis\oasishw

05/03/2010  02:24 PM            18,130 KS0713LCD.c
               3 File(s)         24,461 bytes

 Directory of C:\Projects\OASIS_Review\oasis\Object

12/14/1999  08:36 AM             1,944 SystemStartup.cpp

 Directory of C:\Projects\OASIS_Review\oasis\Object

11/11/1999  05:56 PM               342 SystemStartup.h
               2 File(s)          2,286 bytes

 Directory of C:\Projects\OASIS_Review\oasis\PlaybackManager

05/03/2010  02:24 PM            20,446 CMP3InStream.cpp
05/03/2010  02:24 PM            12,740 CPlaybackManager.cpp

 Directory of C:\Projects\OASIS_Review\oasis\PlaybackManager

05/03/2010  02:24 PM             7,468 CAudioStream.h
05/03/2010  02:24 PM             3,528 CMP3InStream.h
05/03/2010  02:24 PM             4,628 CPlaybackManager.h
05/03/2010  02:24 PM             2,463 SalvageBuffer.h
               6 File(s)         51,273 bytes

 Directory of C:\Projects\OASIS_Review\oasis\QServices

09/27/1999  01:19 PM             1,135 CObjectLock.cpp
05/03/2010  02:24 PM             8,802 QServices.cpp

 Directory of C:\Projects\OASIS_Review\oasis\QServices

10/06/1999  07:14 AM             1,122 CObjectLock.h
05/03/2010  02:24 PM             4,863 QServices.h
               4 File(s)         15,922 bytes

 Directory of C:\Projects\OASIS_Review\oasis\SysMgr

05/03/2010  02:24 PM             3,497 CButtonScanner.cpp
05/03/2010  02:24 PM             3,900 CSysManager.cpp
12/14/1999  08:36 AM             1,944 SystemStartup.cpp

 Directory of C:\Projects\OASIS_Review\oasis\SysMgr

05/03/2010  02:24 PM             1,130 CButtonScanner.h
```

```
05/03/2010  02:24 PM             1,659 CSysManager.h
11/11/1999  05:56 PM               342 SystemStartup.h
               6 File(s)         12,472 bytes

 Directory of C:\Projects\OASIS_Review\oasis\TI5402\Bootloader

05/03/2010  02:24 PM             4,262 buffers.h
05/03/2010  02:24 PM             8,615 regset.h
05/03/2010  02:24 PM               468 stddefs.h
05/03/2010  02:24 PM             1,899 sysproto.h
05/03/2010  02:24 PM               322 Tms320.h

 Directory of C:\Projects\OASIS_Review\oasis\TI5402\Bootloader

12/13/1999  02:03 PM             6,417 DSPMaind.c
12/02/1999  03:56 PM            16,796 RegSetd.C
               7 File(s)         38,779 bytes

 Directory of C:\Projects\OASIS_Review\oasis\TI5402\Maincode

05/03/2010  02:24 PM             4,073 buffers.h
05/03/2010  02:24 PM             8,731 RegSet.h
05/03/2010  02:24 PM               468 stddefs.h
05/03/2010  02:24 PM             1,899 SYSPROTO.H
12/02/1999  04:01 PM               289 Tms320.h

 Directory of C:\Projects\OASIS_Review\oasis\TI5402\Maincode

05/03/2010  02:24 PM             6,247 DSPMain.C
05/03/2010  02:24 PM             8,868 Play.C
05/03/2010  02:24 PM            16,921 RegSet.C
               8 File(s)         47,496 bytes

 Directory of C:\Projects\OASIS_Review\oasis\usb

05/03/2010  02:24 PM            11,067 CUsbManager.cpp

 Directory of C:\Projects\OASIS_Review\oasis\usb

12/13/1999  04:54 PM             1,212 chap_9.h
05/03/2010  02:24 PM             5,121 CUsbManager.h
05/03/2010  02:24 PM             3,734 D12CI.H
05/03/2010  02:24 PM                75 D12ISR.h
05/03/2010  02:24 PM               656 EPPHAL.H
05/03/2010  02:24 PM             6,566 mainloop.h
12/13/1999  05:02 PM               842 protodma.h
05/03/2010  02:24 PM             2,359 usb.h
05/03/2010  02:24 PM             8,512 usb100.h
12/13/1999  05:02 PM             1,035 usbconfig.h

 Directory of C:\Projects\OASIS_Review\oasis\usb

05/03/2010  02:24 PM            10,768 chap_9.c
05/03/2010  02:24 PM             5,634 D12CI.C
05/03/2010  02:24 PM             9,990 D12ISR.C
05/03/2010  02:24 PM               924 EPPHAL.C
05/03/2010  02:24 PM            35,989 usb.c
              16 File(s)        104,484 bytes

 Directory of C:\Projects\OASIS_Review\oasis\VccEmulator

05/03/2010  02:24 PM             3,234 GraphLib.cpp
05/03/2010  02:24 PM             2,060 Oasim.cpp
05/03/2010  02:24 PM             9,458 OasimDlg.cpp
11/27/1999  10:17 AM               207 StdAfx.cpp
```

```
 Directory of C:\Projects\OASIS_Review\oasis\VccEmulator

11/27/1999  10:17 AM             2,245 GraphLib.h
11/27/1999  10:17 AM             1,313 Oasim.h
11/27/1999  10:17 AM             2,035 OasimDlg.h
11/27/1999  10:17 AM             1,735 resource.h
11/27/1999  10:17 AM             1,128 StdAfx.h
               9 File(s)         23,415 bytes

 Directory of C:\Projects\OASIS_Review\oasis\winsim

05/03/2010  02:24 PM             2,390 CButtonSim.cpp
05/03/2010  02:24 PM             5,767 CIdeSim.cpp
05/03/2010  02:24 PM               762 CObject.cpp
05/03/2010  02:24 PM             8,371 CTask.cpp
11/14/1999  12:00 PM             1,364 Dllassert.cpp
05/03/2010  02:24 PM             5,560 FrontPanelOne.cpp
11/14/1999  12:00 PM               439 GrWinDriver.cpp
05/03/2010  02:24 PM             3,841 MPegSim.cpp

 Directory of C:\Projects\OASIS_Review\oasis\winsim

11/14/1999  12:00 PM               208 CButtonSim.h
05/03/2010  02:24 PM             2,854 CIdeSim.h
05/03/2010  02:24 PM             1,997 FrontPanelOne.h
11/14/1999  12:00 PM               300 GrWinDriver.h
11/14/1999  12:00 PM               202 MPegSim.h

 Directory of C:\Projects\OASIS_Review\oasis\winsim

11/14/1999  12:00 PM             9,514 GrDriver.c
              14 File(s)         43,569 bytes

     Total Files Listed:
             193 File(s)      4,381,309 bytes
               0 Dir(s)  67,707,580,416 bytes free
```

# EXHIBIT
# B

```
//--------------------------------------------------------------------------
//
// File:
//              LibraryTree.cpp
//
// Author:
//              Howard Egan
//
// Creation Date:
//              4-Dec-1999
//
// Copyright (c) 1999 E-mu Systems Inc.
// All rights reserved.
//
// Revision History:
//              4-Dec-1999 Original
//
// Module Description:
//              Class CLibTreeMgr creates and makes modifications as necessary
//              to a tree structure whose purpose is to make the track data on disk
//              logically and quickly navigable.  It is currently rebuilt every time
//              the system boots.
//
//              Actual traversal of the tree is via the tree itself.  The root
//              and nodes are implemented as CLibElementNode objects and the leafs
//              are implemented as CLibTrack elements.
//
//
//
//--------------------------------------------------------------------------

#include "LibraryTree.h"
#include "CAttributes.h"
#include "FileSystem.h"
#include "LibElement.h"
#include <assert.h>
#include "CInode.h"
#include <string.h>

//#define Testing
#ifndef Testing
#define TFsFile CFsFile
#else
#include <stdio.h>
#include <dir.h>

class TFsFile{
public:
    TFsFile(FILE *infile);
    void ReadAttributes(CAttributeList *attrList);


    char Album[MAXPATH];
    char Title[MAXPATH];
    char Artist[MAXPATH];
    char Genre[MAXPATH];
    char Playlists[MAXPATH];

};


#endif
```

```
// Eenumerations and strings for categories
// this is a starting point, however before final implementation
// we will probably want a category list on disk that follows the current
// data base.  This along with a overall data base of
// attributes may be useful to prevent versionitis between databases and
// code versions.  The correct approach is not apparent as of
// yet, but this works for the time being.


// more commentary on these strings and enums,,,
// It seems there should be some way to unify them with the actual attribute
// names.  What we are presenting here is just the  displayed categories.

typedef enum _tCatEnum{
    eAlbum,
    eArtist,
    eGenre,
    ePlaylists,
    LengthOfCatEnum
}tCatEnum;

static tBool catItemsAsPlaylists[LengthOfCatEnum] = {
    TRUE, // eAlbum
    FALSE, // eArtist
    FALSE, // eGenre
    TRUE   // ePlayists
};




static const char *catStrings[LengthOfCatEnum] = {
    "ALBUMS",
    "ARTISTS",
    "STYLES",
    "PLAY LISTS"
};

static const char *catDisplayStrings[LengthOfCatEnum] = {
    "ALBUMS",
    "ARTISTS",
    "STYLES",
    "PLAY LISTS"
};

const CLibElementNode *CLibTreeMgr::GetLibRoot(){
    return root;
}

TFsFile *FirstTrack(void);
TFsFile *NextTrack(void);


// Creates the tree from the tracks direcotory
CLibTreeMgr *gpLibTreeMgr = NULL;
//
CLibTreeMgr::CLibTreeMgr(){
    CStr r("root");
    root = new CLibElementNode(r);

    gpLibTreeMgr = this;
```

```
    // add all the categories
    AddCategories();

    // now iterate through each track
    TFsFile *track = FirstTrack();

    while(track){
        // iterates through each category and adds track
        // if applicable
        AddNewTrack(track);
        delete track;
        track = NextTrack();
    }

}

void CLibTreeMgr::AddCategories(void){


    for(int i = 0;i<LengthOfCatEnum;i++){
        const char *catStr = catDisplayStrings[i];
        CStr str(catStr);
        CLibElementNode *nextCatNode = new CLibElementNode(str);

        root->AddChild(nextCatNode);

    }
}



void CLibTreeMgr::AddNewTrack(TFsFile *aTrack){


    // read the Attribute block
    CAttributeList attrList;

    aTrack->ReadAttributes(&attrList);

    // get the track name
    CAttribute *foundTrack = attrList.FindAttribute("TITLE");
    char *foundTrackName;
    if(foundTrack){
        foundTrackName = (char *)foundTrack->GetValue();
    }
    else{
        // we can't support tracks with no name attribute.
        return;
    }


    // get the album attr if it exists.
    CAttribute *foundAlbum = attrList.FindAttribute(catStrings[eAlbum]);
    char *foundAlbumName;
    if(foundAlbum){
        foundAlbumName = (char *)foundAlbum->GetValue();
    }
    else{
        // we can support tracks with no ablum name
        foundAlbumName = NULL;
    }

    // iterate through each category
    CLibElementNode *curCatNode = (CLibElementNode *) root->Get1stChild();
    for(int i = 0;i<LengthOfCatEnum;i++){
        // test from GetSibling at end of loop
```

```
        assert(curCatNode);
        // get the attribute value that for this category
        // from the track file
        const char *catStr = catStrings[i];


        // does this track have an attribute of this
        // name
        CAttribute *foundAttr = attrList.FindAttribute(catStr);


        if(foundAttr){
            // then add this track to the category
            AddNewTrackToCategory(curCatNode,
                                  aTrack,
                                  foundAttr,
                                  foundTrackName,
                                  foundAlbumName,
                                  catItemsAsPlaylists[i]
                                  );
        }
        curCatNode = (CLibElementNode *) curCatNode->GetSibling();
    }
}


void CLibTreeMgr::AddNewTrackToCategory(CLibElementNode *aCatNode,
                            TFsFile *aTrack,
                            CAttribute *anAttribute,
                            const char *aTrackName,
                            const char *anAlbum,
                            tBool categoryItemsArePlaylists){

    // Add the category item if it does not exist
    char *attributeValue = (char *)anAttribute->GetValue();
    CStr str(attributeValue);

    // merely returns the found item if it already exists.
    CLibElementNode *targetNode = aCatNode->AddSubNode(str);
    assert(targetNode);

    // if album non-null add the album
    if(anAlbum){
        if(!categoryItemsArePlaylists){
            //
            targetNode = AddAlbumToCategoryItem(targetNode,anAlbum);
            assert(targetNode);
        }
    }

    CStr trkName(aTrackName);
#ifndef Testing
    tINodeAddr *addr = aTrack->inode->GetAddr();
#else
    tINodeAddr laddr;
    tINodeAddr *addr = &laddr;
#endif
    targetNode->AddTrack(trkName,addr);
}



CLibElementNode *CLibTreeMgr::AddAlbumToCategoryItem(
                    CLibElementNode *aCatItemNode,
                    const char *anAlbum){
```

```
    //
    CLibElementNode *retval;

    CStr str(anAlbum);
    retval = aCatItemNode->AddSubNode(str);
    return retval;
}



static FILE *infile;
static CFsDirectory *fsRoot;
static tFsFileInfo fsInfo;
TFsFile *FirstTrack(void){

#ifdef Testing
    infile = fopen("TestTracks","rt");
    TFsFile *retval = new TFsFile(infile);
    return retval;


#else
    TFsFile *retval = NULL;

    fsRoot = FsGetRoot();

    int status = fsRoot->FirstDirEntry(&fsInfo);

    if(status != -1){
        retval = fsRoot->OpenFile(fsInfo.name);
    }
    return retval;
#endif
}

TFsFile *NextTrack(void){

#ifdef Testing
    if(!feof(infile)){
        TFsFile *retval = new TFsFile(infile);
        return retval;
    }
    return NULL;

#else

    TFsFile *retval = NULL;

    int status = fsRoot->NextDirEntry(&fsInfo);

    if(status != -1){
        retval = fsRoot->OpenFile(fsInfo.name);
    }
    return retval;
#endif

}


#ifdef Testing

TFsFile::TFsFile(FILE *infile){
```

```
    // find the first line
    while(!feof(infile)){
        int c = fgetc(infile);
        if(c == '%')
            break;
    }

    int i = 0;
    while(!feof(infile)){
        int c = fgetc(infile);
        if(c == '*')
            break;
        if((c == 0x0d) || (c == 0x0a)){
            c = 0;
        }
        Album[i] = c;
        i++;
    }
    Album[i] = 0;


    i = 0;
    while(!feof(infile)){
        int c = fgetc(infile);
        if(c == '*')
            break;
        if((c == 0x0d) || (c == 0x0a)){
            c = 0;
        }
        Title[i] = c;
        i++;
    }
    Title[i] = 0;


    i = 0;
    while(!feof(infile)){
        int c = fgetc(infile);
        if(c == '*')
            break;
        if((c == 0x0d) || (c == 0x0a)){
            c = 0;
        }
        Artist[i] = c;
        i++;
    }
    Artist[i] = 0;


    i = 0;
    while(!feof(infile)){
        int c = fgetc(infile);
        if(c == '*')
            break;
        if((c == 0x0d) || (c == 0x0a)){
            c = 0;
        }
        Genre[i] = c;
        i++;
    }
    Genre[i] = 0;

    i = 0;
    while(!feof(infile)){
        int c = fgetc(infile);
        if(c == '*')
```

```
            break;
        if((c == 0x0d) || (c == 0x0a)){
            c = 0;
        }
        Playlists[i] = c;
        i++;
    }
    Playlists[i] = 0;

}

void TFsFile::ReadAttributes(CAttributeList *attrList){

    CAttributeList &l = *attrList;

    int len = strlen(Album)+1;
    int ndx = eAlbum;
    CAttribute *album = new CAttribute(catStrings[ndx],
                                       Album,
                                       len,
                                       0,
                                       FALSE);

    len = strlen(Artist)+1;
    ndx ++;
    CAttribute *artist = new CAttribute(catStrings[ndx],
                                       Artist,
                                       len,
                                       0,
                                       FALSE);

    len = strlen(Genre)+1;
    ndx ++;
    CAttribute *genre = new CAttribute(catStrings[ndx],
                                       Genre,
                                       len,
                                       0,
                                       FALSE);



    len = strlen(Playlists)+1;
    ndx ++;
    CAttribute *playlists = new CAttribute(catStrings[ndx],
                                       Playlists,
                                       len,
                                       0,
                                       FALSE);

    len = strlen(Title)+1;
    CAttribute *title = new CAttribute("TITLE",
                                       Title,
                                       len,
                                       0,
                                       FALSE);

    l.AddAttribute(album);
    l.AddAttribute(artist);
    l.AddAttribute(genre);
    l.AddAttribute(playlists);
    l.AddAttribute(title);

}
#endif

//% Titile * Album * Artist * Genre * Playlists * Codec M = mp3, W = Wav
```

```
void CreateAttributes(CAttributeList *aList,const char *txt){

    static char buf[kiFsMaxPath];

    // starting scan
    while(txt[0] != '%')txt++;
    txt++;
    int i = 0;

    int j = 0;
    while(txt[i] != '*'){
        buf[j] = txt[i];
        i++; j++;
    }
    buf[j] = 0; i++;j = 0;

    int len = strlen(buf) +1;
    CAttribute *title = new CAttribute("TITLE",buf,len,0,TRUE);

    while(txt[i] != '*'){
        buf[j] = txt[i];
        i++; j++;
    }
    buf[j] = 0; i++; j = 0;


    int ndx = eAlbum;
    len = strlen(buf) +1;
    CAttribute *album = new CAttribute(catStrings[ndx],
                                       buf,
                                       len,
                                       0,
                                       TRUE);


    while(txt[i] != '*'){
        buf[j] = txt[i];
        i++; j++;
    }
    buf[j] = 0; i++; j = 0;

    ndx = eArtist;
    len = strlen(buf) +1;
    CAttribute *artist;
    artist = new CAttribute(catStrings[ndx],
                                       buf,
                                       len,
                                       0,
                                       TRUE);

    while(txt[i] != '*'){
        buf[j] = txt[i];
        i++; j++;
    }
    buf[j] = 0; i++; j = 0;

    ndx = eGenre;
    len = strlen(buf) +1;
    CAttribute *genre = new CAttribute(catStrings[ndx],
                                       buf,
                                       len,
                                       0,
                                       TRUE);
```

```
    while(txt[i] != '*'){
        buf[j] = txt[i];
        i++; j++;
    }
    buf[j] = 0; i++; j = 0;
    // skip the play list

    while(txt[i] != 0x00){
        buf[j] = txt[i];
        i++; j++;
    }
    buf[j] = 0; i++; j = 0;

    len = strlen(buf) +1;
    CAttribute *codec = new CAttribute("CODEC",
                                       buf,
                                       len,
                                       0,
                                       TRUE);


    CAttributeList &l = *aList;

    l.AddAttribute(album);
    l.AddAttribute(artist);
    l.AddAttribute(genre);
    l.AddAttribute(codec);
    l.AddAttribute(title);


}
```

# EXHIBIT
# C

```
//---------------------------------------------------------------------------
//
// File:
//              CLibraryTree.h
//
// Author:
//              Howard Egan
//
// Creation Date:
//              4-Dec-1999
//
// Copyright (c) 1999 E-mu Systems Inc.
// All rights reserved.
//
// Revision History:
//              4-Dec-1999 Original
//
// Module Description:
//              Class CLibTreeMgr creates and makes modifications as necessary
//              to a tree structure whose purpose is to make the track data on disk
//              logically and quickly navigable.  It is currently rebuilt every time
//              the system boots.
//
//              Actual traversal of the tree is via the tree itself.  The root
//              and nodes are implemented as CLibElementNode objects and the leafs
//              are implemented as CLibTrack elements.
//
//
//
//---------------------------------------------------------------------------
//---------------------------------------------------------------------------
#ifndef CLibraryTreeH
#define CLibraryTreeH
//---------------------------------------------------------------------------
#include "ProjectTypes.H"
class CLibElement;
class CLibElementTrack;
class CLibElementNode;
class CFsFile;
class CAttribute;

//#define Testing
#ifndef Testing
#define TFsFile CFsFile
#else
class TFsFile;
#endif

class CLibTreeMgr{

public:

    CLibTreeMgr();

    const CLibElementNode *GetLibRoot();

    void AddNewTrack(TFsFile *aTrack);

protected:

    void AddCategories(void);

    void AddNewTrackToCategory(CLibElementNode *aCatNode,
                               TFsFile *aTrack,
                               CAttribute *anAttribute,
                               const char *aTrackName,
```

```
                                const char *anAlbum,
                                tBool categoryItemsArePlaylists);


    CLibElementNode *AddAlbumToCategoryItem(CLibElementNode *aCatItemNode,
                                const char *anAlbum);

    CLibElementNode *root;

};

extern CLibTreeMgr *gpLibTreeMgr;

#endif
```

# EXHIBIT D

```
// LCDLibPage.cpp: implementation of the CLCDLibPage class.
//
//////////////////////////////////////////////////////////////////////

#include "StdInclude.h"

#include "LCDLibPage.h"
#include "LibraryTree.h"
#include "LibElement.h"
#include "LCDMgr.h"

CLCDLibPage::CLCDLibPage()
{
    GDRect rect(0,0,LCDWIDTH-1,LCDHEIGHT-LCDLINEHEIGHT);
    CLCDListBox *lb = new CLCDListBox(rect);
    SetListBox(lb);
    SetSKLabels("Open","Close","Queue");
    cStrListNode=NULL;
    cFirstDisplayedLine=cHighlightedLine=0;

}

void CLCDLibPage::Activate()
{
    setMasterNode(NULL);

    CLCDPage::Activate();
}

void CLCDLibPage::setMasterNode(CLibElementNode *pNode)
{

    cStrListNode=pNode;

    if (cStrListNode)
    {
        CStr aStr;
        cStrListNode->GetStrWithIcon(&aStr);
        GetListBox()->SetTitle(aStr.CharPtr());
    }
    else{
        GetListBox()->SetTitle("MUSIC LIBRARY CATEGORIES");

        cStrListNode = (CLibElementNode *)gpLibTreeMgr->GetLibRoot();
    }

    BuildStrList(cStrListNode);

//  SYS.GetLibraryMgr()->BuildStrList(cStrListNode,GetListBox());
//  GetListBox()->SetFirstDisplayedLine(cStrListNode->GetFirstDisplayedLine());
//  GetListBox()->SetHighlightedLine(cStrListNode->GetHighlightedLine());
    Update();

}

CLibElementNode *CLCDLibPage::GetCurrentPlayableNode(void){

    void *aVal= cListBox->GetHighlightedStrData();
    CLibElementNode *aElem=(CLibElementNode *)aVal;

    if(!aElem){
        return NULL;
    }

    if(aElem->IsPlayableList()){
        return aElem;
```

```
    }
    return NULL;
}

void CLCDLibPage::Update()
{

    static CLibElement *sLastElem=NULL;

    // aCurElem==NULL if there are no children for node
    CLibElement *aCurElem=(CLibElement *)cListBox->GetHighlightedStrData();

    if (IsPageDirty() || (sLastElem!=aCurElem))
    {
        sLastElem=aCurElem;

        // fix menu
        if (!cStrListNode) // at head of library
        {
            // set pos in list
            cFirstDisplayedLine=GetListBox()->GetFirstDisplayedLine();
            cHighlightedLine=GetListBox()->GetHighlightedLine();

            // set open softkey button
            setSK1Enabled(TRUE);
            SetSKLabel1("Open");
        }
        else
        {
            // set pos in list
            cStrListNode->SaveFirstAndHighlightedLine(cListBox->GetFirstDisplayedLine(),
cListBox->GetHighlightedLine());

            // if not at main level, you can open if this node has children and those
    children are nodes
            CLibElement *aChildren=cStrListNode->GetlstChild();

            if (!aChildren)
            {
                setSK1Enabled(FALSE);
            }
            else
            {
                setSK1Enabled(TRUE);
                if (aChildren->GetLibElementType()==kLETNode)
                {
                    SetSKLabel1("Open");
                }
                else
                {
                    SetSKLabel1("Details");
                }
            }
        }

        tBool aQueueable=TRUE;

        if (aCurElem) // there is at least 1 child
        {
            switch (aCurElem->GetLibElementType())
            {
            case kLETNode:
                aQueueable=((CLibElementNode *)aCurElem)->IsPlayableList();
                break;
            case kLETTrack:
                aQueueable=TRUE;
```

```
                break;
            default:
                aQueueable=FALSE;
                ASSERT(FALSE);
                break;
            }
        }
        else
        {
            // no children, thus, nothing to queue
            aQueueable=FALSE;
        }

        setSK2Enabled(cStrListNode!=NULL);
        setSK3Enabled(aQueueable);
        // redraw menu & listbox
        CLCDPage::Draw();
    }

    CLCDPage::Update();

}

void CLCDLibPage::Softkey1Handler(tBool pDown)
{
    // open

    void *aVal=cListBox->GetHighlightedStrData();
    CLibElement *aElem=(CLibElement *)aVal;

    ASSERT(aElem);

    if (aElem->GetLibElementType()==kLETNode)
    {
        GDEnableUpdate(FALSE);
        setMasterNode((CLibElementNode *)aElem);
        GDEnableUpdate(TRUE);
    }
    else
    {
//      ASSERT(aElem->GetLibElementType()==kLETTrack);
//      SYS.GetLCDMgr()->GetTrackInfoPage()->SetTrack((CLibElementTrack *)aElem);
//      SYS.GetLCDMgr()->SetLCDMode(kLMTrackInfo);
    }

}

void CLCDLibPage::Softkey2Handler(tBool pDown)
{
    // close


    if (!cStrListNode) // already at top
        return;

    CLibElement *aElem=cStrListNode->GetParent();

    if ((!aElem/*head of library*/) || (aElem->GetLibElementType()==kLETNode))
    {
        GDEnableUpdate(FALSE);
        setMasterNode((CLibElementNode *)aElem);
        GDEnableUpdate(TRUE);
    }
    else
    {
        ASSERT(FALSE); // can't go up to a track
```

```
    }

}

void CLCDLibPage::Softkey3Handler(tBool pDown)
{
    if(cSK3Enabled){
        void *aVal=cListBox->GetHighlightedStrData();
        CLibElementNode *aElem=(CLibElementNode *)aVal;
        if(aElem){
            gpLcdMgr->HandleQ(aElem);
        }
    }
}
```

# EXHIBIT
# E

```
#ifndef LCDLibPage
#define LCDLibPage

#include "LCDPage.h"

class CLibElementNode; // forward decl

class CLCDLibPage : public CLCDPage
{
public:
    CLCDLibPage();

    virtual void Update();

    virtual void Activate();

    CLibElementNode *GetCurrentPlayableNode(void);

    virtual void Softkey1Handler(tBool pDown);
    virtual void Softkey2Handler(tBool pDown);
    virtual void Softkey3Handler(tBool pDown);
private:
    void setMasterNode(CLibElementNode *pNode);

    // where we are in root list (since there's no CLibElementNode for it)
    int cFirstDisplayedLine;
    int cHighlightedLine;

    CLibElementNode *cStrListNode; // node that is being displayed on this page
};

#endif // #ifndef LCDLibPage
```

# EXHIBIT
# F

```
// LCDMgr.cpp: implementation of the CLCDMgr class.
//
//////////////////////////////////////////////////////////////////////

#include "StdInclude.h"


#include "LCDMgr.h"

#include "GrCustom.h"
#include "CStr.h"

#include "LCDListBox.h"
#include "CButtonScanner.h"
#include "GrResources.h"
#include "QServices.h"
#include "CPlaybackManager.h"
#include "CDspManager.h"
#include "NowPlayingQ.h"
//////////////////////////////////////////////////////////////////////
// Construction/Destruction
//////////////////////////////////////////////////////////////////////

CLCDMgr::CLCDMgr()
{
    cLCDMode=kLMMax;
    cIsModal=FALSE;
    cSplashAge = 0;

    // eIdRawButtonEvents
    rawButtonInput = new CInputPort(CPort::eIdRawButtonEvents);

    // set the max/min watermarks for the port
    rawButtonInput->SetThrottle(100,0);

    // we also need a wakeup event for signaling, so we simply get the next
    // available event flag, but this is not pretty...

    // get the currently used flags
    int CurrentlyUsed = CEventFlag::GetUsedFlags();

    // look for the next available flag, sizeof(tEventFlag) << 3 will give us the
    // number of flags
    for (int i = 0; i < (sizeof(tEventFlag) << 3); i++){
        if (!(CurrentlyUsed & (1 << i))){
            // found an unused flag
            wakeupEvent = new CEventFlag(i);
            break;
        }
    }

    // return immediately if error
    if (wakeupEvent == NULL){
        valid = false;
        return;
    }

    rawButtonInput->SetWakeupEvent(wakeupEvent);

}

CLCDMgr::~CLCDMgr()
{

}
```

```
// kickoff the task
void CLCDMgr::StartTask(void){

    Schedule("LCD Manager",  6, 1000, 0);
}

void CLCDMgr::Start(){
    // initialization stuff goes here
    Initialize();

    // ongoing thread execution here in this loop
    CMessage *btnMsg;
    do{
        btnMsg = rawButtonInput->Get();
        while(btnMsg){
            unsigned long btn = btnMsg->GetLParam();
            ButtonLogic(btn);
            delete btnMsg;
            TimerTicked();
            btnMsg = rawButtonInput->Get();
        }

        Sleep(5);
        if(cLCDNextPage != cLCDMode){
            SetLCDMode(cLCDNextPage);
        }
    }while(1);

}




void CLCDMgr::Initialize()
{
    SetLCDMode(kLMSplash);
}

void mDrawSplash(tCoord pY)
{
    GDEnableUpdate(FALSE);
//  GDClearCanvas();
    GDCopyPixmap(&splash_PICT,0,-pY,FALSE);
    GDEnableUpdate(TRUE);
}


void CLCDMgr::SetNextPage(tLCDMode pLCDMode){
    cLCDNextPage = pLCDMode;
}

void CLCDMgr::SetLCDMode(tLCDMode pLCDMode)
{
    cLCDNextPage = cLCDMode=pLCDMode;
    cCurPage=NULL;

    switch (cLCDMode)
    {
    case kLMSplash:
        mDrawSplash(0);
//      cCurPage=&cLCDSplashPage;
        SetLCDMode(kLMLibrary);
        break;
```

```
    case kLMMenu:
        cCurPage=&cLCDMenuPage;
        break;
    case kLMLibrary:
        cCurPage=&cLCDLibPage;
        break;
    case kLMTrackInfo:
        cCurPage=&cLCDTrackInfoPage;
        break;
    case kLMListening:
        cCurPage=&cLCDListeningPage;
        break;
    case kLMRecord:
        break;
    case kLMQList:
        cCurPage=&cLCDQListPage;
        break;
    default:
        ASSERT(FALSE);
        return;
    }

    if (cCurPage)
        cCurPage->Activate();
}
#ifdef FancySplash
extern "C" int OasisAnimationCount;
extern "C" void *OasisAnimation[];
#else
int OasisAnimationCount = 1;
void *OasisAnimation[1] = {
&splash_PICT
};
#endif
void CLCDMgr::TimerTicked()
{


    switch (cLCDMode)
    {
    case kLMListening:
        break;
    case kLMSplash:
#define SPLASHTIME (10)
        cSplashAge++;
        if(cSplashAge < OasisAnimationCount){
            GDEnableUpdate(FALSE);
            GDCopyPixmap((GrPixmap*)OasisAnimation[cSplashAge],0,0,FALSE);
            GDEnableUpdate(TRUE);
        }

        if (cSplashAge>SPLASHTIME)
        {
            tCoord aY=(cSplashAge-SPLASHTIME)*4;
            if (aY<LCDHEIGHT)
            {
                GDEnableUpdate(FALSE);
                GDCopyPixmap((GrPixmap*)OasisAnimation[OasisAnimationCount -1],0,aY,
    FALSE);
                GDEnableUpdate(TRUE);
//              mDrawSplash(aY);
            }
            else
            {
                GDEnableUpdate(FALSE);
                GDClearCanvas();
```

```
                GDEnableUpdate(TRUE);
                SetLCDMode(kLMLibrary);
            }
        }
    }

}


void CLCDMgr::ShowTimedMessage(CStr &pMsg, tDeciseconds pDecisecs)
{
#define BORDER (10)
    cTimedMessageCountdown=pDecisecs;
    GDEnableUpdate(FALSE);

    GDSetFont(defaultBoldFont);

    tCoord aWidth=GDGetStringWidth(pMsg.CharPtr());
    tCoord aHeight=GDGetStringHeight(pMsg.CharPtr());

    if (aWidth>LCDWIDTH)
    {
        GDSetFont(tinyFont);
        aWidth=GDGetStringWidth(pMsg.CharPtr());
        aHeight=GDGetStringHeight(pMsg.CharPtr());
    }

    GDSetFillPattern(&grWhite);
    GDSetPenPattern(&grBlack);
    GrRect aMsgRegion={
        (LCDWIDTH-aWidth)/2-BORDER,
        (LCDHEIGHT-aHeight)/2-BORDER,
        (LCDWIDTH+aWidth)/2+BORDER,
        (LCDHEIGHT+aHeight)/2+BORDER};
    GDFillRect(&aMsgRegion);
    GDDrawRect(&aMsgRegion);

    GDDrawString(
        (LCDWIDTH-aWidth)/2,
        (LCDHEIGHT-aHeight)/2,
        pMsg.CharPtr());

    GDEnableUpdate(TRUE);
}




void CLCDMgr::ButtonLogic(unsigned long aBtn){

    CButtonScanner::tOasisButton btn = (CButtonScanner::tOasisButton)aBtn;


    if(btn == CButtonScanner::eNone){
        return;
    }

    // dispatch for handling by the current screen
    if(btn == CButtonScanner::eF1Btn){
        cCurPage->SoftkeyHandler(1,TRUE);
        return;
    }
    if(btn == CButtonScanner::eF2Btn){
        cCurPage->SoftkeyHandler(2,TRUE);
        return;
```

```
    }

    if(btn == CButtonScanner::eF3Btn){
        cCurPage->SoftkeyHandler(3,TRUE);
        return;
    }

    if(btn == CButtonScanner::eUpBtn){
        cCurPage->ScrollUp();
        return;
    }
    if(btn == CButtonScanner::eDnBtn){
        cCurPage->ScrollDown();
        return;
    }

    // the remainder of the buttons are handled directly by the
    // lcdMgr and beyond

    if(btn == CButtonScanner::ePlayBtn){
        handlePlayButton();
        return;
    }

    if(btn == CButtonScanner::eLibBtn){
        cCurPage->DeActivate();
        if(cLCDMode != kLMLibrary){
            SetNextPage(kLMLibrary);
        }
        else{
            SetNextPage(kLMListening);
        }
    }
}

static CDspManager::tPlaybackState currentPlayState = CDspManager::eStateStop;


void CLCDMgr::handlePlayButton(void){
    // on play you will want to get the currently
    // selected filename from libList and then to a
    //

    CLibElementNode *aNode;
    if(!cLCDLibPage.IsActive()){
        // then set the selected node to NULL;
        Lock();
        pCurrentLibElementNode = NULL;
        UnLock();
    }
    else{
        Lock();
        pCurrentLibElementNode = cLCDLibPage.GetCurrentPlayableNode();
        UnLock();
    }

    // we should be posting the button press to the system manager at
    // this point, but we are just defering the logic to here for now.

    // if there is something to add to the q list then do that
    // don't need to protect on read in this thread cause this is the only
    // modifying code.
    if(pCurrentLibElementNode){
        gpNowPlayingManager->PlayElement((CLibElement *)pCurrentLibElementNode);
    }
```

```
    // this should be done on a scheduled basis but for now we just set it
    // directly
    SetNextPage(kLMListening);

//    gpPlaybackManager->PlayPause();

}
void CLCDMgr::HandleQ(CLibElementNode *anElem){

    if(anElem){
        gpNowPlayingManager->QElement((CLibElement *)anElem);
    }
}
```

# EXHIBIT
# G

```
// LCDMgr.h: interface for the CLCDMgr class.
//
//////////////////////////////////////////////////////////////////////

#if !defined(AFX_LCDMGR_H__D180AFA3_46AB_11D3_ACB2_0000E83FB110__INCLUDED_)
#define AFX_LCDMGR_H__D180AFA3_46AB_11D3_ACB2_0000E83FB110__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

#include "CTask.h"
#include "CObjectLock.h"
#include "LCDTrackInfoPage.h"
#include "LCDMenuPage.h"
#include "LCDLibPage.h"
#include "LCDListeningPage.h"
#include "LCDQListPage.h"
//#include "LCDFunctionsPage.h"
//#include "LCDGamesPage.h"
#include "LCDSplashPage.h"

class CLCDListBox; // forward decl
class CLibElementNode; // forward decl
class CLibElementTrack; // forward decl
class CInputPort;


class CLCDMgr:public CTask, public CObjectLock {

public:
    CLCDMgr();
    virtual ~CLCDMgr();



    void Initialize();


    void StartTask(void);

    void SetNextPage(tLCDMode pLCDMode);

    // Feedback from lib manager screen.
    // Q's are qualified by lib mangager
    // screen, than passed on with an element
    void HandleQ(CLibElementNode *anElem);

protected:
    void handlePlayButton(void);

    void ButtonLogic(unsigned long aBtn);
    void TimerTicked();

    void SetLCDMode(tLCDMode pLCDMode);
    tLCDMode GetLCDMode() {return cLCDMode;};

    // show a temporary message to user over rest of LCD
    void ShowTimedMessage(CStr &pMsg,tDeciseconds pDecisecs);

    CLCDPage *GetCurPage() {return cCurPage;};
    CLCDTrackInfoPage *GetTrackInfoPage() {return &cLCDTrackInfoPage;};
    CLCDMenuPage *GetMenuPage() {return &cLCDMenuPage;};
    CLCDLibPage *GetLibPage() {return &cLCDLibPage;};
    CLCDListeningPage *GetListeningPage() {return &cLCDListeningPage;};
    CLCDQListPage *GetQListPage() {return &cLCDQListPage;};
```

```
    CLCDPage *cCurPage;

    CLCDSplashPage cLCDSplashPage;
    CLCDListeningPage cLCDListeningPage;
    CLCDQListPage cLCDQListPage;
    CLCDMenuPage cLCDMenuPage;
    CLCDLibPage cLCDLibPage;
//  CLCDFunctionsPage cLCDFunctionsPage;
    CLCDTrackInfoPage cLCDTrackInfoPage;
//  CLCDGamesPage cLCDGamesPage;

    tLCDMode cLCDMode,cLCDNextPage;
    tBool cIsModal;
    int cSplashAge;
    tDeciseconds cTimedMessageCountdown;


    // actual thread function
    virtual void Start();

    // port to communicate with the LCDMgr
    // eIdRawButtonEvents
    CInputPort *rawButtonInput;

    // maintained for use by Play transport key
    // when we were in the lib screen and play
    // was pressed.  Otherwise it must remain NULL;
    CLibElementNode *pCurrentLibElementNode;

    CEventFlag *wakeupEvent;
    tBool valid;


};

extern CLCDMgr *gpLcdMgr;

#endif // !defined(AFX_LCDMGR_H__D180AFA3_46AB_11D3_ACB2_0000E83FB110__INCLUDED_)
```

# EXHIBIT
# H

```
//////////////////////////////////////////////////////////////////////
//
// TransportMgr.cpp: implementation of the CTransportMgr class.
//
//////////////////////////////////////////////////////////////////////

#include "StdInclude.h"
#include "LibElement.h"
#include <assert.h>
#include "NowPlayingQ.h"

CNowPlayingMgr::~CNowPlayingMgr()
{
    CNowPlayingList *aDel;

    while (cHead)
    {
        aDel=cHead;
        cHead=cHead->cNext;
        delete aDel;
    }
}

// remove an element from QList
void CNowPlayingMgr::RemoveElement(CNowPlayingList *pRemQueueEntry)
{
    CNowPlayingList *aRem=cHead,*aPrev=NULL;
    while (aRem)
    {
        if (aRem==pRemQueueEntry)
        {
            if (aPrev)
            {
                // are we removing head?
                if(aRem == cTail){
                    cTail = aPrev;
                    cTail->cNext = NULL;
                }
                else{
                    aPrev->cNext=aRem->cNext;
                }
                delete aRem;
                return;
            }
            else
            {//
                if(aRem == cHead){
                    // also the tail
                    if(aRem == cTail){
                        cHead = cTail = NULL;
                        delete aRem;
                        return;
                    }
                    // just the head
                    cHead = aRem->cNext;
                    delete aRem;
                    return;
                }
            }
        }
        aPrev=aRem;
        aRem=aRem->cNext;
    }
}

void CNowPlayingMgr::PlayElement(CLibElement *pElement){
```

```
     PrependElement(pElement);
     GotoFirstElement();
}

void CNowPlayingMgr::QElement(CLibElement *pElement){
    AppendElement(pElement);
    // anything there now?
    if(!cCurElement){
        GotoFirstElement();
    }
}

void CNowPlayingMgr::ClearQList()
{
    CNowPlayingList *aTmp;
    while (cHead!=NULL)
    {
        aTmp=cHead;
        cHead=cHead->cNext;
        delete aTmp;
    }
    cCurElement=NULL;
    cCurTrack=NULL;
}

// also serves as GotoFirstTrack() (same function)
void CNowPlayingMgr::GotoFirstElement()
{
    cCurElement=cHead;
    if (cCurElement)
    {
        switch (cCurElement->cElement->GetLibElementType())
        {
        case kLETTrack:
            cCurTrack=cCurElement->cElementAsTrack;
            break;
        case kLETNode:
            cCurTrack=(CLibElementTrack *)cCurElement->cElementAsNode->GetlstChild();
            assert(cCurTrack->GetLibElementType()==kLETTrack);
            break;
        default:
            assert(FALSE);
            break;
        }
    }
    else
        cCurTrack=NULL;
}

void CNowPlayingMgr::GotoNextElement()
{
    if (!cCurElement)
        cCurElement=cHead;
    if (!cCurElement)
    {
        cCurTrack=NULL;
        return;
    }

    // get next element
    cCurElement=cCurElement->cNext;
    if (!cCurElement)
    {
        // no more elements
        cCurTrack=NULL;
        return;
```

```
    }

    switch (cCurElement->cElement->GetLibElementType())
    {
    case kLETNode:
        cCurTrack=(CLibElementTrack *)cCurElement->cElementAsNode->GetIstChild();
        assert(cCurTrack->GetLibElementType()==kLETTrack);
        return;
    case kLETTrack:
        cCurTrack=cCurElement->cElementAsTrack;
        return;
    default:
        assert(FALSE);
        return;
    }
}

// if prev element is a node, last track in node (not first) becomes current
void CNowPlayingMgr::GotoPrevElement()
{
    if (!cCurElement)
    {
        cCurTrack=NULL;
        return;
    }

    // get prev element
    cCurElement=cCurElement->cPrev;
    if (!cCurElement)
    {
        // no more elements
        cCurTrack=NULL;
        return;
    }

    switch (cCurElement->cElement->GetLibElementType())
    {
    case kLETNode:
        cCurTrack=(CLibElementTrack *)cCurElement->cElementAsNode->GetLastChild();
        assert(cCurTrack->GetLibElementType()==kLETTrack);
        return;
    case kLETTrack:
        cCurTrack=cCurElement->cElementAsTrack;
        return;
    default:
        assert(FALSE);
        return;
    }
}

void CNowPlayingMgr::GotoNextTrack()
{
    if (!cCurElement)
        cCurElement=cHead;
    if (!cCurElement)
    {
        cCurTrack=NULL;
        return;
    }

    switch (cCurElement->cElement->GetLibElementType())
    {
    case kLETNode:
        cCurTrack=(CLibElementTrack *)cCurTrack->GetSibling();
        if (cCurTrack)
            return; // got next track... done
```

```
        // go to next element
        GotoNextElement();
        return;
    case kLETTrack:
        GotoNextElement();
        return;
    default:
        assert(FALSE);
        return;
    }
}

void CNowPlayingMgr::GotoPrevTrack()
{
    if (!cCurElement)
    {
        cCurTrack=NULL;
        return;
    }

    switch (cCurElement->cElement->GetLibElementType())
    {
    case kLETNode:
        cCurTrack=(CLibElementTrack *)cCurTrack->GetPrevSibling();
        if (cCurTrack)
            return; // got next track... done

        // go to next element
        GotoPrevElement();
        break;
    case kLETTrack:
        GotoPrevElement();
        break;
    default:
        assert(FALSE);
        break;
    }
    if (!cCurElement)
    {
        GotoFirstElement();
    }
}

// how many elements in QList
int CNowPlayingMgr::GetElemCount()
{
    int aCount=0;
    CNowPlayingList *aIdx=cHead;
    while (aIdx)
    {
        aCount++;
        aIdx=aIdx->cNext;
    }
    return aCount;
}

// returns # of tracks in QList
int CNowPlayingMgr::GetTrackCount()
{
    return getTrackIndex(NULL);
}

// returns 1-based index, or 0 if not found
int CNowPlayingMgr::GetPlayingTracksIndex()
{
```

```
    return getTrackIndex(GetCurTrack());
}

// returns index in queue of pTrack (1-based), or # of tracks if pTrack==NULL, or 0 if
    pTrack!=NULL but isn't in list
int CNowPlayingMgr::getTrackIndex(CLibElementTrack *pTrack)
{
    CNowPlayingList *aElem=cHead;
    CLibElementTrack *aIdx;

    int aCount=0;

    while (aElem)
    {
        switch (aElem->cElement->GetLibElementType())
        {
        case kLETNode:
            aIdx=(CLibElementTrack *)aElem->cElementAsNode->Get1stChild();
            while (aIdx)
            {
                assert(aIdx->GetLibElementType()==kLETTrack);
                aCount++;
                if (pTrack==aIdx)
                    return aCount; // match found, return
                aIdx=(CLibElementTrack *)aIdx->GetSibling();
            }
            break;
        case kLETTrack:
            aCount++;
            if (aElem->cElementAsTrack==pTrack)
                return aCount;
            break;
        }
        aElem=aElem->cNext;
    }

    if (pTrack)
        return 0; // return NOT-FOUND
    else
        return aCount; // return # of tracks total in list
}

void CNowPlayingMgr::PrependElement(CLibElement *pElement)
{
    cSublistHead=cSublistTail=NULL;
    createTrackSublist(pElement);
    if (cSublistTail) // if we got anything back from this node to prepend
    {
        // if anything exists yet, glue it in
        if (cHead)
        {
            cSublistTail->cNext=cHead;
            cHead->cPrev=cSublistTail;
            cHead=cSublistHead;
        }
        else
        {
            // else, just make track list be sublist
            cHead=cSublistHead;
            cTail=cSublistTail;
        }
    }
}

void CNowPlayingMgr::AppendElement(CLibElement *pElement)
{
```

```
    cSublistHead=cSublistTail=NULL;
    createTrackSublist(pElement);
    if (cSublistTail) // if we got anything back from this node to prepend
    {
        // if anything exists yet, glue it in
        if (cHead)
        {
            cSublistHead->cPrev=cTail;
            cTail->cNext=cSublistHead;
            cTail=cSublistTail;
        }
        else
        {
            // else, just make track list be sublist
            cHead=cSublistHead;
            cTail=cSublistTail;
        }
    }
}

void CNowPlayingMgr::createTrackSublist(CLibElement *pElement)
{
    cSublistHead=new CNowPlayingList;
    cSublistTail=cSublistHead;
    cSublistHead->cElement=pElement;

    return;
#ifdef Bling
    // in jukebox mode, cueing a track uses JUST that track
    if ((SYS.GetTransportMgr()->GetTransportMode()==kTMJukebox) && (pElement->
    GetLibElementType()==kLETTrack))
    {
        cSublistHead=new CNowPlayingList;
        cSublistTail=cSublistHead;
        cSublistHead->cElementAsTrack=(CLibElementTrack *)pElement;
    }
    else
        createTrackSublist_recurse(pElement);
#endif
}

void CNowPlayingMgr::createTrackSublist_recurse(CLibElement *pElement)
{
    if (pElement->GetLibElementType()==kLETNode)
        pElement=((CLibElementNode *)pElement)->Get1stChild();

    while (pElement)
    {
        switch (pElement->GetLibElementType())
        {
        case kLETNode:
            createTrackSublist(pElement);
            break;
        case kLETTrack:
            if (cSublistHead)
            {
                cSublistTail->cNext=new CNowPlayingList;
                cSublistTail->cNext->cPrev=cSublistTail;
                cSublistTail=cSublistTail->cNext;
            }
            else
            {
                cSublistHead=new CNowPlayingList;
                cSublistTail=cSublistHead;
            }
            cSublistTail->cElementAsTrack=((CLibElementTrack *)pElement);
```

```
            break;
        default:
            assert(FALSE);
            break;
        }
        pElement=pElement->GetSibling();
    }
}
```

# EXHIBIT
# I

```
// QMgr.h: interface for the CNowPlayingMgr class.
//
//////////////////////////////////////////////////////////////////////

#ifndef NOWPLAYINGQ__HH
#define NOWPLAYINGQ__HH
#include "CObjectLock.h"
class CLibElement; // forward decl
class CLibElementNode; // forward decl
class CLibElementTrack; // forward decl

class CNowPlayingList:public CObjectLock
{
public:
    CNowPlayingList() {cElement=NULL;cPrev=cNext=NULL;};

    union
    {
        CLibElement *cElement;
        CLibElementNode *cElementAsNode;
        CLibElementTrack *cElementAsTrack;
    };

    CNowPlayingList *cPrev;
    CNowPlayingList *cNext;
};

class CNowPlayingMgr:public CObjectLock
{
public:
    CNowPlayingMgr() {cHead=cTail=NULL;cCurTrack=NULL;cCurElement=NULL;};
    virtual ~CNowPlayingMgr();

    void PrependElement(CLibElement *pElement);
    void AppendElement(CLibElement *pElement);

    void PlayElement(CLibElement *pElement);
    void QElement(CLibElement *pElement);

    // get ptr to currently playing element
    CNowPlayingList *GetCurElement() {return cCurElement;};
    CLibElementTrack *GetCurTrack() {return cCurTrack;};

    // clear all tracks out of QList
    void ClearQList();

    // remove an element from QList
    void RemoveElement(CNowPlayingList *pRemElem);

    // position cur element to be first/next/prev element
    void GotoFirstElement();
    void GotoNextElement();
    void GotoPrevElement();

    // how many elements in QList
    int GetElemCount();

    // position to next/prev track (not element)
    void GotoNextTrack();
    void GotoPrevTrack();

    // how many total tracks in all elements in Qlist
    int GetTrackCount();
    // returns 1-based index, or 0 if not found
    int GetPlayingTracksIndex();
```

```
private:
    // returns index in queue of pTrack (1-based), or # of tracks if pTrack==NULL, or 0 ↙
    if pTrack!=NULL but isn't in list
    int getTrackIndex(CLibElementTrack *pTrack);

    void createTrackSublist(CLibElement *pElement);
    void createTrackSublist_recurse(CLibElement *pElement);

    CNowPlayingList *cHead,*cTail;
    CNowPlayingList *cCurElement; // pointer to current track within linked list of ↙
    playing tracks
    CLibElementTrack *cCurTrack; // current track within cCurElement (==cCurElement if ↙
    cCurElement is a CLibElementTrack)
    CNowPlayingList *cSublistHead,*cSublistTail;
};
extern CNowPlayingMgr *gpNowPlayingManager;
#endif // !defined(AFX_QUEUEMGR_H__33193520_4C24_11D3_ACB2_0000E83FB110__INCLUDED_)
```

# EXHIBIT
# J

```
// LCDQListPage.cpp: implementation of the CLCDQListPage class.
//
//////////////////////////////////////////////////////////////////////


#include "StdInclude.h"


#include "LCDQListPage.h"
#include "NowPlayingQ.h"
#include "CDspManager.h"
#include "LibElement.h"

CLCDQListPage::CLCDQListPage()
{
    GDRect rect(0,0,LCDWIDTH-1,LCDHEIGHT-LCDLINEHEIGHT);
    CLCDListBox *lb = new CLCDListBox(rect);
    SetListBox(lb);


    SetSKLabels("NowPlaying","Remove","---");
    GetListBox()->SetEmptyListStr("<nothing else waiting to play>");

    setSoftkeyNames();
}

// returns if something was updated
void CLCDQListPage::Update()
{
    // update no more than once per second
    static int sUpdateCount=0;
    if (!IsPageDirty())
    {
        // if the page isn't dirty, then only update once per second
        if (sUpdateCount++>10)
            sUpdateCount=0;
        else
            return;
    }

    static void *sLastElem=NULL;
    static int sLastElemCount=0;
    static CDspManager::tPlaybackState lastPlaybackState = CDspManager::eStatePlay;

    CDspManager::tPlaybackState currentPlaybackState = gpDspManager->GetPlayState();

    tBool aExpandLists=FALSE; // should we list album as album or as individual tracks

    if (aExpandLists)
    {
        int aCurElemCount= gpNowPlayingManager->GetTrackCount();
        ASSERT(FALSE); // unsupported
    }
    else
    {
        int aCurElemCount= gpNowPlayingManager->GetElemCount();
        CNowPlayingList *aCurElem= gpNowPlayingManager->GetCurElement();

        if ((aCurElem!=sLastElem) || (aCurElemCount!=sLastElemCount) ||
            (currentPlaybackState != lastPlaybackState) || IsPageDirty())
        {
            sLastElem=aCurElem;
            sLastElemCount=aCurElemCount;

            GetListBox()->DeleteStrings();
```

```
            if( (currentPlaybackState != CDspManager::eStateStop) && aCurElem)
            {
                CStr aNowPlaying("Playing: "),aElemStr;

                aCurElem->cElement->GetStrWithIcon(&aElemStr);

                aNowPlaying.StrCat(aElemStr);

                GetListBox()->SetTitle(&aNowPlaying);

                aCurElem=aCurElem->cNext;

                while (aCurElem)
                {
                    aCurElem->cElement->GetStrWithIcon(&aElemStr);
                    GetListBox()->AddString(&aElemStr,aCurElem);
                    aCurElem=aCurElem->cNext;
                }
            }
            else
                GetListBox()->SetTitle("Nothing Playing");
            CLCDPage::Draw();
        }
    }

    CLCDPage::Update();
}

void CLCDQListPage::setSoftkeyNames()
{
#ifdef Blinq
    CStr aLabel;

    switch (SYS.State().GetPlayMode())
    {
    default:
        ASSERT(FALSE);
    case kPMPlayOnce:
        aLabel.StrCpy("PlayOnce");
        break;
    case kPMRepeat:
        aLabel.StrCpy("Repeat");
        break;
    case kPMRandom:
        aLabel.StrCpy("Random");
        break;
    }
#endif
}

void CLCDQListPage::Softkey1Handler(tBool pDown)
{

    SetNextPage(kLMListening);

}

// remove from queue
void CLCDQListPage::Softkey2Handler(tBool pDown)
{
    CNowPlayingList *aRemQueueEntry=(CNowPlayingList *)cListBox->GetHighlightedStrData();
    if(aRemQueueEntry){
        gpNowPlayingManager->RemoveElement(aRemQueueEntry);
        // now we need to rebuild the contents
        cPageDirty=TRUE;
        Update();
```

```
    }

}

void CLCDQListPage::Softkey3Handler(tBool pDown)
{
#ifdef Bling
    // toggle thru play modes
    long aMode=SYS.State().GetPlayMode();
    aMode++;
    if (aMode==kPMMax)
        aMode=0;
    SYS.State().SetPlayMode((tPlayMode)aMode);
#endif
}

void CLCDQListPage::Activate()
{
    CLCDPage::Activate();
}
```

# EXHIBIT
# K

```
#ifndef LCDQListPage
#define LCDQListPage

#include "LCDPage.h"

class CLCDQListPage : public CLCDPage
{
public:
    CLCDQListPage();

    virtual void Update();

    virtual void Softkey1Handler(tBool pDown);
    virtual void Softkey2Handler(tBool pDown);
    virtual void Softkey3Handler(tBool pDown);

    virtual void Activate();

private:
    void setSoftkeyNames();
};

#endif // #ifndef LCDQListPage
```

# EXHIBIT
# L

```
// LCDListBox.cpp: implementation of the CLCDListBox class.
//
//////////////////////////////////////////////////////////////////////

#include "StdInclude.h"

#include "LCDListBox.h"
#include "LCDMgr.h"
#include "CStr.h"
#include "GrResources.h"

//////////////////////////////////////////////////////////////////////
// Construction/Destruction
//////////////////////////////////////////////////////////////////////

#define SCROLLBARWIDTH (8)
#define SCROLLBARBUTTONWIDTH (SCROLLBARWIDTH-2)
#define SCROLLBARBUTTONHEIGHT (16)
#define TEXTRIGHT ((cNumItemsInList>cNumLinesDisplayed)?cRect.right-SCROLLBARWIDTH:cRect.right)
#define LISTBOX_INDENT (4)


/* this special drawstring draws an icon in the first position */
#include <string.h>
void GDDrawStringWithIcon(short pX,short pY,char *pString)
{
    int aIdx;
    char aTmp;
    char aCopy[128],*aString=aCopy;
    strcpy(aCopy,pString);
    while (*aString)
    {
        for (aIdx=0;(aString[aIdx]>ICON_MAX);aIdx++)
            ;
        /* if it we accrued any characters > ICON_MAX, draw them */
        if (aIdx)
        {
            aTmp=aString[aIdx];
            aString[aIdx]=0;
            GDDrawString(pX,pY,aString);
            pX+=GDGetStringWidth(aString);
            aString=&aString[aIdx];
            *aString=aTmp;
        }
        /* if the end of that str was an icon, draw it */
        if (*aString) /* found an icon */
        {
            switch (*aString)
            {
            case ICON_LIST:
                GDCopyPixmap(&kIconList,pX,pY,0);
                pX+=kIconList.dimension.w;
                break;
            case ICON_NODE:
                GDCopyPixmap(&kIconNode,pX,pY,0);
                pX+=kIconNode.dimension.w;
                break;
            case ICON_SONG:
                GDCopyPixmap(&kIconTrack,pX,pY,0);
                pX+=kIconTrack.dimension.w;
                break;
            case ICON_VOICE:
                GDCopyPixmap(&kIconVoice,pX,pY,0);
                pX+=kIconVoice.dimension.w;
```

```
                break;
            default:
                break;
            }
            aString++;
        }
    }
}


    CStrListAssoc *cStrListHead,*cStrListTail;


CLCDListBox::CLCDListBox(GDRect &pRect)
{
    cStrListHead=cStrListTail=NULL;
    cRect=pRect;
    cNumLinesDisplayed=(cRect.Height()+1)/LCDLINEHEIGHT-1; // -1 for space for heading
    cTitle.StrCpy("Generic List");
    cEmptyListStr.StrCpy("<empty>");
    cHighlightedLine=0;
    cFirstDisplayedLine=0;
}

CLCDListBox::~CLCDListBox()
{
    CStrListAssoc *aDel;
    while (cStrListHead)
    {
        aDel=cStrListHead;
        cStrListHead=cStrListHead->cNext;
        delete aDel->cStr;
        delete aDel;
    }
}

void CLCDListBox::invertLine(int pLine)
{
    // pLine refers to the item, but the line we want to invert is pLine+1
    // because of the title line. Thus...
    pLine++;

    GDEnableUpdate(FALSE);
    GrTransferMode aOldMode=GDGetTransferMode();
    GDSetTransferMode(GR_XFER_XOR);
    GDSetFillPattern(&grBlack);
    GrRect aInvertRect={cRect.left,pLine*LCDLINEHEIGHT,TEXTRIGHT,pLine*LCDLINEHEIGHT+
    LCDLINEHEIGHT-1};
    GDFillRect(&aInvertRect);
    GDSetTransferMode(aOldMode);
    GDEnableUpdate(TRUE);
}

void CLCDListBox::SetHighlightedLine(int pHighlightedLine)
{
    cHighlightedLine=pHighlightedLine;
    Draw();
}

void CLCDListBox::SetFirstDisplayedLine(int pFirstDisplayedLine)
{
    cFirstDisplayedLine=pFirstDisplayedLine;
    Draw();
}
```

```
void CLCDListBox::ScrollDn()
{
    GDEnableUpdate(FALSE);
    // if we're highlighting a line other than the bottom line...
    if (ibetween(0,cHighlightedLine,cNumLinesDisplayed-2))
    {
        // if there are more items to scroll down to...
        if (ibetween(0,cHighlightedLine,cNumItemsInList-2))
        {
            invertLine(cHighlightedLine);
            cHighlightedLine++;
            invertLine(cHighlightedLine);
        }
    }
    else
    {
        if (cFirstDisplayedLine<cNumItemsInList)
        {
            cFirstDisplayedLine++;
            Draw();
        }
    }
    drawScrollbar();
    GDEnableUpdate(TRUE);
}

void CLCDListBox::ScrollUp()
{
    GDEnableUpdate(FALSE);
    if (ibetween(1,cHighlightedLine,cNumLinesDisplayed-1))
    {
        invertLine(cHighlightedLine);
        cHighlightedLine--;
        invertLine(cHighlightedLine);
    }
    else
    {
        if (cFirstDisplayedLine>0)
        {
            cFirstDisplayedLine--;
            Draw();
        }
    }
    drawScrollbar();
    GDEnableUpdate(TRUE);
}

void CLCDListBox::countNumItemsInList()
{
    CStrListAssoc *aStrs=cStrListHead;
    cNumItemsInList=0;
    while (aStrs!=NULL)
    {
        cNumItemsInList++;
        aStrs=aStrs->cNext;
    }

    // if there aren't enough lines to fill display and highlight is beyond filled area, ↙
    move it up
    if (cHighlightedLine>=cNumItemsInList)
        cHighlightedLine=cNumItemsInList-1;

    // make sure first displayed line makes sense
    cFirstDisplayedLine=ibound(0,cFirstDisplayedLine, imax(0,cNumItemsInList- ↙
    cNumLinesDisplayed) );
```

```
}

CStrListAssoc *CLCDListBox::getHighlightedPtr()
{
    int aLine=GetHighlightedIndex();

    CStrListAssoc *aStrs=cStrListHead;
    if (!aStrs)
        return NULL;
    for (int aIdx=0;aIdx<aLine;aIdx++)
    {
        aStrs=aStrs->cNext;
        if (!aStrs)
        {
            ASSERT(FALSE);
            return NULL;
        }
    }
    if (aStrs)
        return aStrs;
    else
        return NULL;
}

void *CLCDListBox::GetHighlightedStrData()
{
    CStrListAssoc *aPtr=getHighlightedPtr();
    if (aPtr)
        return aPtr->cData;
    else
        return NULL;
}

CStr *CLCDListBox::GetHighlightedStr()
{
    CStrListAssoc *aPtr=getHighlightedPtr();
    if (aPtr)
        return aPtr->cStr;
    else
        return NULL;
}

void CLCDListBox::DeleteStrings()
{
    CStrListAssoc *aTmp;

    while (cStrListHead)
    {
        aTmp=cStrListHead;
        cStrListHead=cStrListHead->cNext;
        delete aTmp->cStr;
        delete aTmp;
    }
    cStrListHead=NULL;
    cStrListTail=NULL;
}

void CLCDListBox::AddString(CStr *pStr,void *pPtr)
{
    if (cStrListTail)
    {
        cStrListTail->cNext=new CStrListAssoc;
        cStrListTail=cStrListTail->cNext;
    }
    else
    {
```

```
        cStrListTail=new CStrListAssoc;
        cStrListHead=cStrListTail;
    }
    cStrListTail->cStr=new CStr(*pStr);
    cStrListTail->cData=pPtr;
    cStrListTail->cNext=NULL;
}

void CLCDListBox::drawScrollbar()
{
    if (cNumItemsInList>cNumLinesDisplayed)
    {
        GDEnableUpdate(FALSE);
        GDSetFillPattern(&grWhite);
        GrRect aInnerRegion={TEXTRIGHT+1+2,cRect.top+1,cRect.right-1,cRect.bottom-1};
        GDFillRect(&aInnerRegion);
        GDSetPenPattern(&grBlack);
        GrRect aOuterBorder={TEXTRIGHT+1,cRect.top,cRect.right,cRect.bottom};
        GDDrawRect(&aOuterBorder);
        tCoord aY=1+ // 1 line below top of scroll
            (cRect.Height()-2-SCROLLBARBUTTONHEIGHT+1)* // number of positions
    scrollbarbutton can be in
            (cFirstDisplayedLine)/(cNumItemsInList-cNumLinesDisplayed); // % down the
    scrollbarbutton should be
//      DRAWBITMAP(TEXTRIGHT+2,
//          ibound(1,aY,cRect.bottom-1-SCROLLBARBUTTONHEIGHT),IDB_SCROLLBARBUTTON);
        GDSetFillPattern(&grGray);
        GrRect aButtonRegion={TEXTRIGHT+2,aY,TEXTRIGHT+8,aY+14};
        GDFillRect(&aButtonRegion);
        GDEnableUpdate(TRUE);
    }
}

void CLCDListBox::Draw()
{
    countNumItemsInList();

    CStrListAssoc *aStrs=cStrListHead;
    int aIdx;
    for (aIdx=0;aIdx<cFirstDisplayedLine;aIdx++)
    {
        if (aStrs==NULL)
        {
            ASSERT(FALSE);
            break;
        }
        aStrs=aStrs->cNext;
    }

    tCoord aY=cRect.top;
    GDEnableUpdate(FALSE);

    // erase entire area of listbox
    GrRect aListboxRegion={cRect.left,cRect.top,cRect.right,cRect.bottom};
    GDSetFillPattern(&grWhite);
    GDFillRect(&aListboxRegion);

    // draw heading
    GDSetPenPattern(&grWhite);
    GDSetFont(tinyFont);
    GDDrawStringWithIcon(cRect.left,aY,
        cTitle.CharPtr());
    aY+=LCDLINEHEIGHT;

    // draw strings
    if (aStrs)
```

```
    {
        for (aIdx=0;aIdx<cNumLinesDisplayed;aIdx++)
        {
            if (!aStrs) // if we're out of strings, stop
                break;

            GDDrawStringWithIcon(cRect.left+LISTBOX_INDENT,aY,
                aStrs->cStr->CharPtr());

            if (aIdx==cHighlightedLine)
                invertLine(cHighlightedLine);

            aY+=LCDLINEHEIGHT;

            aStrs=aStrs->cNext;
        }
    }
    else
        GDDrawStringWithIcon(cRect.left+LISTBOX_INDENT,aY,
            cEmptyListStr.CharPtr());

    // draw scroll location indicator
    drawScrollbar();

    GDEnableUpdate(TRUE);
}
```

# EXHIBIT
# M

```
// LCDListBox.h: interface for the CLCDListBox class.
//
//////////////////////////////////////////////////////////////////////

#ifndef __LCDListBox_h__
#define __LCDListBox_h__

#include "CStr.h"
#include "GrCustom.h"


// this is moved from GrCustom to here
// because GrCustom is sometimes compiled
// strictly as extern C.

class GDRect : public GrRect
{
public:
    GDRect()
    {
        left=0;
        top=0;
        right=0;
        bottom=0;
    };
    GDRect(short pLeft,short pTop,short pRight,short pBottom)
    {
        left=pLeft;
        top=pTop;
        right=pRight;
        bottom=pBottom;
    };
    short Height() {return  bottom-top;};
};


class CLCDListBox
{
public:
    CLCDListBox(GDRect &pRect);
    virtual ~CLCDListBox();

    void DeleteStrings();
    void AddString(CStr *pStr,void *pPtr);

    void Draw();

    void ScrollDn();
    void ScrollUp();

//  CStrListAssoc *&GetStrListHead() {return cStrListHead;};
//  CStrListAssoc *&GetStrListTail() {return cStrListTail;};

    void *GetHighlightedStrData();
    CStr *GetHighlightedStr();

    void SetTitle(const char *pTitle) {cTitle.StrCpy(pTitle);};
    void SetTitle(CStr *pTitle) {SetTitle(pTitle->CharPtr());};

    // store string to show when list is empty
    void SetEmptyListStr(char *pEmptyListStr) {cEmptyListStr.StrCpy(pEmptyListStr);};

    void SetHighlightedLine(int pHighlightedLine);
    int GetHighlightedLine() {return cHighlightedLine;};

    // which item is highlighted (between 0 and
```

```
    int GetHighlightedIndex() {return GetFirstDisplayedLine()+GetHighlightedLine();};

    void SetFirstDisplayedLine(int pFirstDisplayedLine);
    int GetFirstDisplayedLine() {return cFirstDisplayedLine;};

private:
    void invertLine(int pLine);
    void countNumItemsInList();
    void drawScrollbar();
    CStrListAssoc *getHighlightedPtr();

    int cNumLinesDisplayed; // how many lines are displayed at once on display
    int cHighlightedLine; // which line is highlighted (from 0 to cNumLinesDisplayed-1)
    int cNumItemsInList; // how many items are in strlist
    int cFirstDisplayedLine; // which of cNumItemsInList lines is at top of display
    CStr cTitle; // title/heading for listbox
    CStr cEmptyListStr; // string to display in an empty list

    GDRect cRect;

    CStrListAssoc *cStrListHead,*cStrListTail;
};

#endif // !defined(AFX_LCDListBox_H__BD813660_4A74_11D3_ACB2_0000E83FB110__INCLUDED_)
```

# EXHIBIT N

```
// LCDPage.cpp: implementation of the CLCDPage class.
//
//////////////////////////////////////////////////////////////////////

#include "StdInclude.h"
#include "LibElement.h"
#include <assert.h>

#ifdef _DEBUG
#undef THIS_FILE
static char THIS_FILE[]=__FILE__;
#define new DEBUG_NEW
#endif

#include "LCDPage.h"
#include "LCDMgr.h"

void CLCDPage::Draw()
{
    GDEnableUpdate(FALSE);
    if (cListBox)
        cListBox->Draw();
    if (cMenuEnabled)
        drawMenu();
    GDEnableUpdate(TRUE);
}
void CLCDPage::SetNextPage(tLCDMode pLCDMode){
    gpLcdMgr->SetNextPage(pLCDMode);
}


void CLCDPage::drawMenu()
{
    GDEnableUpdate(FALSE);
    GDSetFillPattern(&grWhite);
    GrRect aMenuRegion=(0,LCDHEIGHT-LCDLINEHEIGHT,LCDWIDTH,LCDHEIGHT-1);
    GDFillRect(&aMenuRegion);
    GDSetPenPattern(&grBlack);
    GDSetFont(tinyFont);
    char *aStr;

    if (cSK1Enabled)
        aStr= cSKLabel1.CharPtr();
    else
        aStr= "---";

    GDDrawString(0,LCDHEIGHT-LCDLINEHEIGHT,aStr);

    if (cSK2Enabled)
        aStr= cSKLabel2.CharPtr();
    else
        aStr= "---";

    GDDrawString((LCDWIDTH-GDGetStringWidth(aStr))/2,
        LCDHEIGHT-LCDLINEHEIGHT,aStr);


    if (cSK3Enabled)
        aStr= cSKLabel3.CharPtr();
    else
        aStr= "---";

    GDDrawString(LCDWIDTH-GDGetStringWidth(aStr),
        LCDHEIGHT-LCDLINEHEIGHT,aStr);

    GDEnableUpdate(TRUE);
```

```
}

CLCDPage::~CLCDPage()
{
    if (cListBox)
        delete cListBox;
}

void CLCDPage::SoftkeyHandler(int pSoftkeyNum,tBool pDown)
{
    switch (pSoftkeyNum)
    {
    case 1:
        if (cSK1Enabled)
            Softkey1Handler(pDown);
        break;
    case 2:
        if (cSK2Enabled)
            Softkey2Handler(pDown);
        break;
    case 3:
        if (cSK3Enabled)
            Softkey3Handler(pDown);
        break;
    default:
        ASSERT(FALSE);
        break;
    }
}

void CLCDPage::SetPageDirty()
{
    GDEnableUpdate(FALSE);
    GDClearCanvas();
    GDEnableUpdate(TRUE);

    cPageDirty=TRUE;
}

void CLCDPage::Activate()
{
    SetPageDirty();
    Update();
    Draw();
    isActive = TRUE;
}

void CLCDPage::DeActivate()
{
    isActive = FALSE;
}




void CLCDPage::addStr(CLibElementNode *pElement)
{
    CStr aStr;
    CStr *pStr = &aStr;

    pElement->GetStrWithIcon(pStr);

    cListBox->AddString(pStr, pElement);

#if 0
```

```
    if (cListBox->GetStrListHead()==NULL)
    {
        cListBox->GetStrListHead()=new CStrListAssoc;
        cListBox->GetStrListHead()->cStr=pStr;
        cListBox->GetStrListHead()->cData=pElement;
        cListBox->GetStrListHead()->cNext=NULL;
        cListBox->GetStrListTail()=cListBox->GetStrListHead();
    }
    else
    {
        cListBox->GetStrListTail()->cNext=new CStrListAssoc;
        cListBox->GetStrListTail()=cListBox->GetStrListTail()->cNext;
        cListBox->GetStrListTail()->cStr=pStr;
        cListBox->GetStrListTail()->cData=pElement;
    }
#endif

}


void CLCDPage::BuildStrList(CLibElementNode *pNode){

    if(!cListBox)
        return;

    assert(pNode);

    if(!pNode)
        return;

    cListBox->DeleteStrings();

    CLibElementNode *aElem= (CLibElementNode *) pNode->GetlstChild();

    while (aElem){
        addStr(aElem);
        aElem = (CLibElementNode *) aElem->GetSibling();
    }
}


void CLCDPage::ScrollUp(void){
    if(cListBox)
        cListBox->ScrollUp();
}

void CLCDPage::ScrollDown(void){
    if(cListBox)
        cListBox->ScrollDn();
}


void CLCDPage::BuildStrList(void *pNode){;}

void CLCDPage::Softkey1Handler(tBool pDown) {};
void CLCDPage::Softkey2Handler(tBool pDown) {};
void CLCDPage::Softkey3Handler(tBool pDown) {};
```

# EXHIBIT
# O

```
#ifndef LCDPage
#define LCDPage

#include "LCDListBox.h"
#include "GeneralTypes.h"
//#include "SoftkeyActions.h"
class CLibElementNode;
extern "C" {
    #include "GrCustom.h"
};

#define LCDWIDTH (SCREEN_WIDTH-4) // real LCD width is 132, but Graphics wants a multiple
     of 8
#define LCDHEIGHT (SCREEN_HEIGHT)
#define LCDNUMTEXTLINES (8)
#define LCDLINEHEIGHT (LCDHEIGHT/LCDNUMTEXTLINES)
#define LCDMEDLINEHEIGHT (LCDHEIGHT/6)
#define LCDLRGLINEHEIGHT (LCDHEIGHT/4)

enum tLCDMode
{
    kLMSplash,
    kLMMenu,
    kLMLibrary,
    kLMTrackInfo,
    kLMListening,
    kLMQList,
    kLMRecord,
    kLMGames,
    kLMMax
};


class CLCDPage
{
public:
    CLCDPage()
    {
        cMenuEnabled=TRUE;
        cSK1Enabled=cSK2Enabled=cSK3Enabled=TRUE;
        cListBox=NULL;
        cShowMenu=TRUE;
    };

    virtual void Draw();

    virtual ~CLCDPage();

    virtual void Activate();
    virtual void DeActivate();

    void SetNextPage(tLCDMode pLCDMode);

    void SetListBox(CLCDListBox *pListBox) {cListBox=pListBox;};
    CLCDListBox *GetListBox() {return cListBox;};


    void BuildStrList(CLibElementNode *anElement);

    void SetShowMenu(tBool pShowMenu) {cShowMenu=pShowMenu;};
    virtual void BuildStrList(void *pNode);//{;}

    virtual void Softkey1Handler(tBool pDown);// {};
    virtual void Softkey2Handler(tBool pDown);// {};
    virtual void Softkey3Handler(tBool pDown);// {};
```

```
    virtual void ScrollUp(void);
    virtual void ScrollDown(void);

    void SetSKLabel1(const char *pLabel) {cSKLabel1.StrCpy(pLabel);drawMenu();};
    void SetSKLabel2(const char *pLabel) {cSKLabel2.StrCpy(pLabel);drawMenu();};
    void SetSKLabel3(const char *pLabel) {cSKLabel3.StrCpy(pLabel);drawMenu();};
    void SetSKLabels(const char *pLabel1,const char *pLabel2,const char *pLabel3)
    {
        cSKLabel1.StrCpy(pLabel1);
        cSKLabel2.StrCpy(pLabel2);
        cSKLabel3.StrCpy(pLabel3);
        drawMenu();
    };

    virtual void Update() {cPageDirty=FALSE;};

    void SetPageDirty();
    tBool IsPageDirty() {return cPageDirty;}; // do we need complete redraw
    tBool IsActive(){return isActive;}
    void SoftkeyHandler(int pSoftkeyNum,tBool pDown);

    void addStr(CLibElementNode *pElement);

protected:
    void setMenuEnabled(tBool pMenuEnabled) {cMenuEnabled=pMenuEnabled;};
    void setSK1Enabled(tBool pEnabled) {cSK1Enabled=pEnabled;};
    void setSK2Enabled(tBool pEnabled) {cSK2Enabled=pEnabled;};
    void setSK3Enabled(tBool pEnabled) {cSK3Enabled=pEnabled;};

    void drawMenu();

    CLCDListBox *cListBox;
    CStr cSKLabel1,cSKLabel2,cSKLabel3;
    tBool cShowMenu;
    tBool cPageDirty; // if TRUE, a complete LCD redraw is needed
    tBool cMenuEnabled; // should menu get displayed
    tBool cSK1Enabled,cSK2Enabled,cSK3Enabled; // is each menu item enabled
    tBool isActive;
};


#endif // #ifndef LCDPage
```

# EXHIBIT
# P

MASTER REDLINE COPY

# Oasis MP3 Audio Player
# Marketing Requirements Document

Revision 0.3

Dan Freeman 3/9/99

## Objectives

1. Expand Creative's product line of MP3 audio products.
2. Differentiate Creative's product from current MP3 players by offering proprietary DSP audio processing features.
3. FCS ~~late Summer~~ October 1999. ⟹ January?

## Product Definition

Oasis is a portable MP3 audio player based on a 2.5" hard disk drive. Its high capacity and portability allows users to take a large collection of music with them virtually wherever they go, without having to carry any bulky CD's or other recorded media along. MP3 songs are downloaded from a host computer via a USB interface. A SmartMedia card interface facilitates exchange of MP3 songs with other portable MP3 players, such as Nomad and Rio. Oasis can also be used as a peripheral disk storage device for a computer.

## Product Positioning

**High-Capacity "Jukebox" Audio Player**

Oasis is a portable jukebox that can store 50 audio CD's worth of music on its internal hard disk. Using their PC and its CD ROM drive, users can easily "record" their audio CDs into the MP3 format and download them to Oasis. Users can also download MP3 encoded audio over the internet. Oasis includes a SmartMedia card slot, so MP3 songs can be exchanged with other MP3 players via ~~SmartMedia cards~~. Songs stored on Oasis can be browsed via the built-in LCD display, and can be programmed to play back in any order.

**Proprietary DSP Audio Processing**

DSP audio processing features implemented in Oasis will include four-channel surround, tone controls, headphones EQ, reverb, and time compression / expansion. These advanced features will set Oasis apart from the current MP3 players available, and will give Creative a competitive edge in the MP3 player market.

wired I/F to Nomad

## Compelling Product Features

**Capacity**

Store a library of 50 audio CD's on a single, portable audio player. The 2.1GB hard disk in Oasis will store approximately 33 hours of 128kbps encoded MP3 song data, or 900 hours of Audible format "spoken word" encoded data.

**Portability**

Oasis can be operated in a car, in a plane, at the beach, or virtually anywhere. Oasis is portable, but not "wearable" like Nomad or Rio. Although the internal hard disk is shock tolerant, Oasis should not subjected to excessive shock or vibration associated with activities like jogging or operating a jackhammer. Oasis can be used anywhere one would expect to be able to use a laptop computer. Oasis is small: approximately 11cm x 9cm x 2.5cm. Oasis can be powered by a DC adapter, or by 4 AA batteries. The operating time under battery power is expected to be about 8 hours.

**DSP Features**

A four-channel "surround sound" feature will be implemented on Oasis. Unlike the "fixed EQ settings" paradigm implemented on the Rio player, Oasis will implement traditional bass and treble controls. A "headphones EQ" feature will be implemented, which precompensates for the frequency response of the speakers in the headphones. ~~A time scaling feature will be provided, which will facilitate cueing through music and voice data.~~ A reverb algorithm will also be implemented, of similar quality to the one available on SoundBlaster Live! This is a large number of DSP features to implement for FCS. It is likely that a subset of these features will be made available for FCS. The remaining DSP features can be made available via software upgrades after FCS.

**Downloadable Firmware**

The software that runs Oasis can be downloaded from a host computer via the USB port. This allows us to offer software upgrades and feature enhancements to Oasis customers on an ongoing basis.

~~SmartMedia Card Slot~~

The SmartMedia slot on Oasis offers a mechanism for interchange of MP3 song data with other MP3 players. The desire is to allow Oasis to upload MP3 songs from a SmartMedia card, play songs from a SmartMedia card, and download songs to a SmartMedia card. The level of interchange that we choose to implement is likely to be dictated by copyright protection issues: It may be necessary to only allow *playing* of songs from SmartMedia, and *moving* (not copying) songs to and from a SmartMedia card.

**Song Search and Playlist Support**

The user interface on Oasis will allow users to browse songs by song name, artist name, or genre. Song playlists can be assembled so that any number of songs can be played in any order. Multiple playlists can be created and stored on the internal hard disk. ID3 Tags version 2.0.

**CDDB "CD Database" Support**

CD Database support will be provided. This allows users to download song and artist information about their CD's from the internet, and store it on Oasis. The song search and browsing features will take advantage of the CDDB data.

**USB Interface to Host PC**

Users download their audio data from their computer to Oasis over the USB port. The maximum download speed will be approximately 500KB/s. At this transfer rate, it will take approximately 80 seconds to download one CD's worth of music (40 minutes) that has already been encoded into the MP3 format.

**Host Software**

A full-featured, well-integrated host application program will ship with Oasis. For audio data manipulation, the host software will include a CD "ripper" which will extract data from Audio CD's, an MP3 encoder, a CD player, an MP3 player. For song cataloging and organization, the host software will implement librarian features, and will include features to access the CDDB over the internet to download audio CD song and artist information. Also, file upload and download to Oasis will be implemented.

**Optional "Wired" Remote Control**

A wired remote control accessory will be offered which allows remote operation of Oasis. Same as Nomad

**Copyright Protection: SDMI Support**

SDMI is the Secure Digital Music Initiative. SDMI's intent is to facilitate copyright protection of digital music, and they plan to establish a standard that multiple vendors can comply with, in time for products that will ship during the Christmas holidays. It may be a strategic decision to comply with the SDMI standard.

**User Interface**

Custom LCD, similar to or the same as used in Nomad II.

Buttons: Play, Stop, etc. The number of buttons and their function is TBD.

**Audio Interfaces**

Four channels of analog audio output: Stereo headphones/line out A, stereo headphones/line out B.

**Digital Interfaces**

USB jack, ~~SmartMedia slot.~~

**Power Supply**

DC input jack, battery compartment.

## Product Features - Post FCS

**"Audible" Format Audio Playback Support**

Sometime after FCS, a software upgrade will be made available that supports the Audible format. The huge storage capacity of Oasis will allow users to store entire audio "books" encoded in the Audible "spoken word" format. The time scaling DSP features will allow users to modify the speed of the audio playback.

**DSP Features**

The downloadable firmware feature of Oasis allows us to make new DSP algorithms available to users in the form of software upgrades.

## Product Features - "Nice to Have"

**Audio Recording**

Using an analog stereo line input, Oasis can record audio data into .WAV files, which can be uploaded to the host computer via the USB port.

Digital optical S/PDIF in with SCMS

## Applications

**Home or Portable Jukebox**
Record 50 of your CD's on Oasis, play them at home or at the office, or take them with you anywhere.

**MP3 Player**
Download MP3 songs from the internet and play them on Oasis.

**"Dance Party DJ" Player**
Record dance tunes, organize them into playlists, and be a DJ at a dance party.

**Audible Book Player**
Download Audible books from the internet, and play them in the car, on the plane, at the beach, etc.

**External HDD Computer Peripheral**
Oasis is a stand-alone external hard disk drive that implements the DOS file system, so it can be used as an external storage device for a laptop computer.

## Competitive Differentiation

**Storage Capacity**
Current MP3 players (Rio, Pontis, MP-Man, Nomad) store <100MB of audio data. Oasis offers 20 times the capacity. Users can store a large CD library on Oasis.

**DSP Features**
Four-channel surround, ~~Time scaling~~, reverb, headphones EQ, tone controls - current MP3 players do not offer these advanced features.

## Target Customers

- Anybody with a CD library and a computer.
- Current MP3 player customers, who want a high-capacity player.
- Future: "Audible Books" customers.

## Delivery Date and Sales Forecast

**FCS**
~~Late Summer 1999~~. October 1999

**Sales Forecast**
TBD.

## Target Cost

ESP of $249 (Web-direct sales).
$299 channel

## Competition MP3 Diskman CAXO (Taiwan)

No competing products are currently shipping. Saehan has advertised their MP-H10 MP3 player, which is also based on a 2.5" hard disk drive. This product is not yet available. See http://www.mpman.com/eng_new/ for details on the Saehan player.

## Marketing Strategy

**Target Customers**
- Anybody with a CD library and a computer.
- Current MP3 player customers, who want a high-capacity player.
- "Audible Books" customers.

**Distribution**
- Direct Internet Sales
- Other Channels (TBD)

**Pricing**
- $249 (Web direct)

$299 channel

**Competition**

None yet....

**Positioning**

Portable jukebox that can store 50 audio CD's worth of music on its internal hard disk. Data interchange with Nomad or other MP3 players via SmartMedia cards.

**Next Product**

Set-top version.

## System and Other Requirements

**Minimum Computer System Requirements**

- Microsoft Windows 98
- 100MHz or higher Pentium
- SVGA graphics adapter (256 colors, 640x480)
- Available USB connection
- 16MB RAM
- 5MB free hard disk space (or more for the bundled software)
- Installed mouse
- Installed CD ROM or DVD ROM drive

**Documentation**

- User's guide

**Accessories**

- USB cable (for PC connection)
- Installation and software CD (for PC connection)

**Localization**

- International English version only

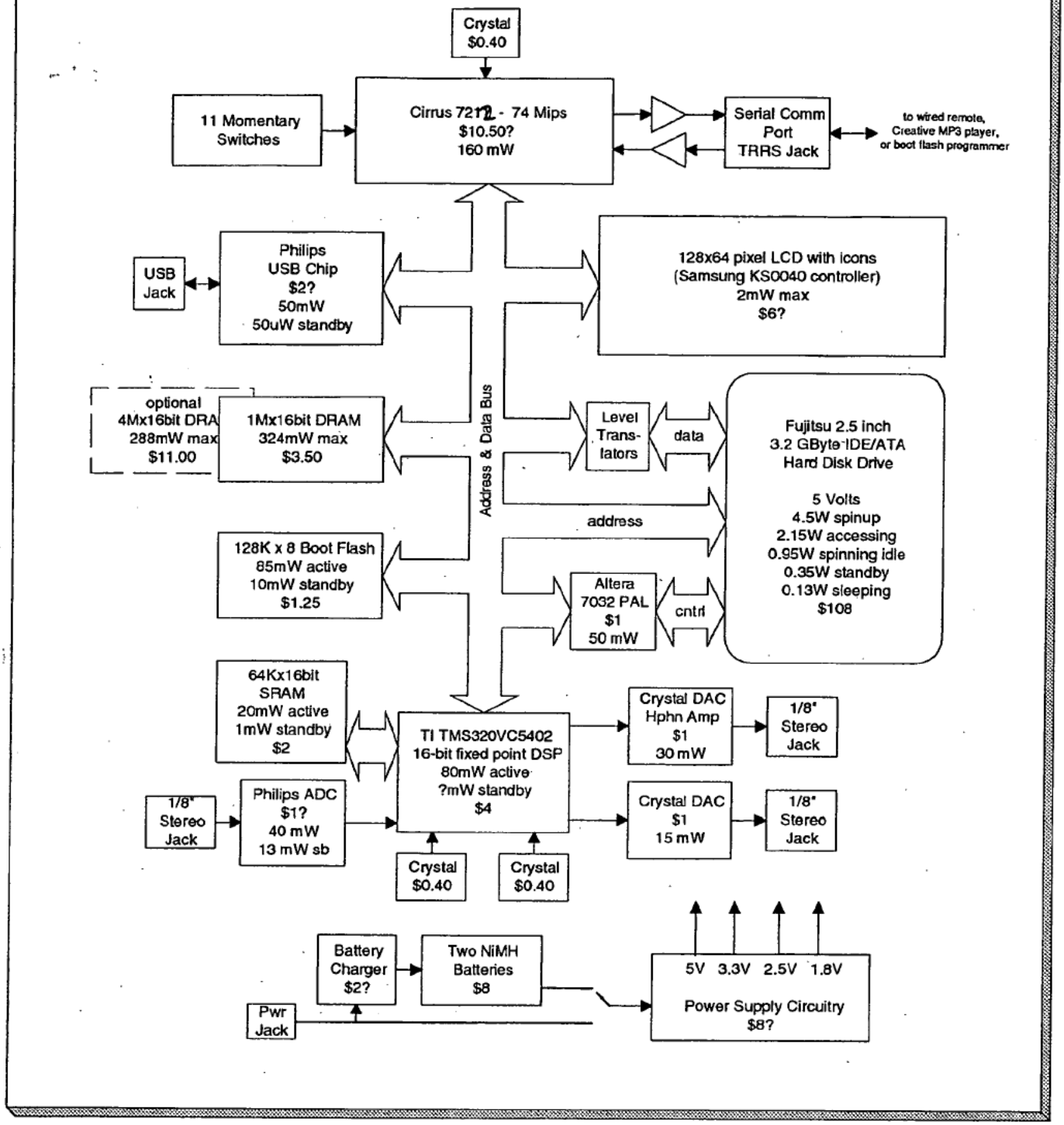
## Revision History

| Revision | Date | By | Changes |
|---|---|---|---|
| 0.3 | 8 March 1999 | Dan Freeman | Add SmartMedia card slot. Add downloadable firmware option. Add two more analog outputs. Delete built-in mic. Accommodate support of Audible format, but don't implement at FCS. Make analog recording feature "nice to have". |
| 0.2 | 1 March 1999 | Dan Freeman | Minor revisions prior to distribution. |
| 0.1 | 28 February 1999 | Dan Freeman | Initial Draft |

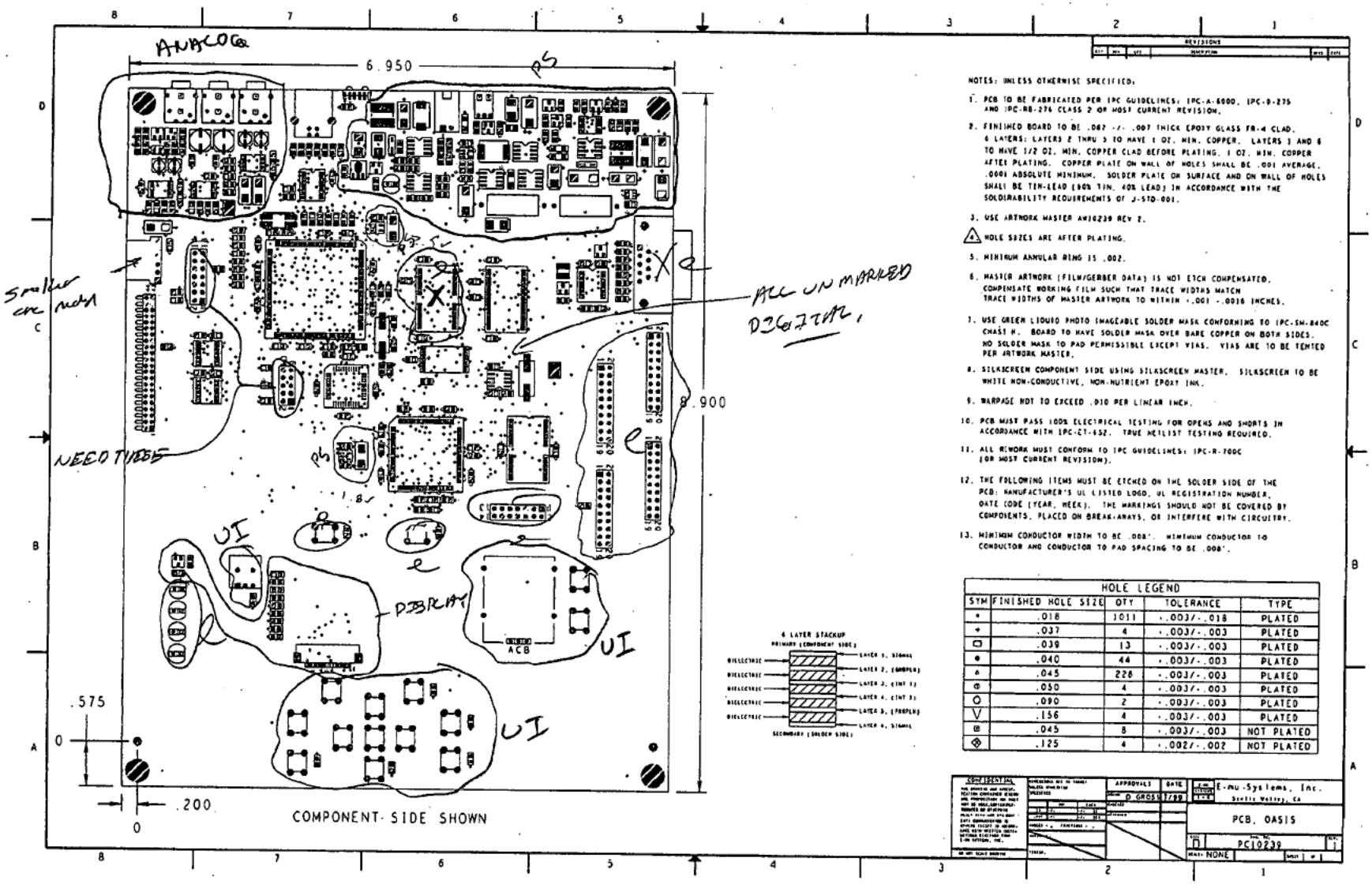
# EXHIBIT
# Q

# Oasis Block Diagram

May 26, 1999



# EXHIBIT
# R




NOTES: UNLESS OTHERWISE SPECIFIED.

1. PCB TO BE FABRICATED PER IPC GUIDELINES: IPC-A-600D, IPC-D-275 AND IPC-RB-276 CLASS 2 OR MOST CURRENT REVISION.
2. FINISHED BOARD TO BE .062 +/- .007 THICK EPOXY GLASS FR-4 CLAD. 6 LAYERS: LAYERS 2 THRU 5 TO HAVE 1 OZ. MIN. COPPER. LAYERS 1 AND 6 TO HAVE 1/2 OZ. MIN. COPPER CLAD BEFORE PLATING. 1 OZ. MIN. COPPER AFTER PLATING. COPPER PLATE ON WALL OF HOLES SHALL BE .001 AVERAGE, .0001 ABSOLUTE MINIMUM. SOLDER PLATE ON SURFACE AND ON WALL OF HOLES SHALL BE TIN-LEAD (60% TIN, 40% LEAD) IN ACCORDANCE WITH THE SOLDERABILITY REQUIREMENTS OF J-STD-001.
3. USE ARTWORK MASTER AM10239 REV 2.
4. HOLE SIZES ARE AFTER PLATING.
5. MINIMUM ANNULAR RING IS .002.
6. MASTER ARTWORK (FILM/GERBER DATA) IS NOT ETCH COMPENSATED. COMPENSATE WORKING FILM SUCH THAT TRACE WIDTHS MATCH TRACE WIDTHS OF MASTER ARTWORK TO WITHIN +.001 -.0016 INCHES.
7. USE GREEN LIQUID PHOTO IMAGEABLE SOLDER MASK CONFORMING TO IPC-SM-840C CHASS H. BOARD TO HAVE SOLDER MASK OVER BARE COPPER ON BOTH SIDES. NO SOLDER MASK TO PAD PERMISSIBLE EXCEPT VIAS. VIAS ARE TO BE TENTED PER ARTWORK MASTER.
8. SILKSCREEN COMPONENT SIDE USING SILKSCREEN MASTER. SILKSCREEN TO BE WHITE NON-CONDUCTIVE, NON-NUTRIENT EPOXY INK.
9. WARPAGE NOT TO EXCEED .010 PER LINEAR INCH.
10. PCB MUST PASS 100% ELECTRICAL TESTING FOR OPENS AND SHORTS IN ACCORDANCE WITH IPC-ET-652. TRUE NETLIST TESTING REQUIRED.
11. ALL REWORK MUST CONFORM TO IPC GUIDELINES: IPC-R-700C (OR MOST CURRENT REVISION).
12. THE FOLLOWING ITEMS MUST BE ETCHED ON THE SOLDER SIDE OF THE PCB: MANUFACTURER'S UL LISTED LOGO, UL REGISTRATION NUMBER, DATE CODE (YEAR, WEEK). THE MARKINGS SHOULD NOT BE COVERED BY COMPONENTS, PLACED ON BREAK-AWAYS, OR INTERFERE WITH CIRCUITRY.
13. MINIMUM CONDUCTOR WIDTH TO BE .008". MINIMUM CONDUCTOR TO CONDUCTOR AND CONDUCTOR TO PAD SPACING TO BE .008".

HOLE LEGEND

| SYM | FINISHED HOLE SIZE | QTY | TOLERANCE | TYPE |
|---|---|---|---|---|
| * | .018 | 1011 | +.003/-.018 | PLATED |
| + | .037 | 4 | +.003/-.003 | PLATED |
| □ | .039 | 13 | +.003/-.003 | PLATED |
| ● | .040 | 44 | +.003/-.003 | PLATED |
| △ | .045 | 228 | +.003/-.003 | PLATED |
| ⊕ | .050 | 4 | +.003/-.003 | PLATED |
| ○ | .090 | 2 | +.003/-.003 | PLATED |
| V | .156 | 4 | +.003/-.003 | PLATED |
| ⊠ | .045 | 8 | +.003/-.003 | NOT PLATED |
| ◈ | .125 | 4 | +.002/-.002 | NOT PLATED |

CONFIDENTIAL

APPROVALS: D. GROSS 7/99

E-mu Systems, Inc. Scotts Valley, CA

PCB, OASIS

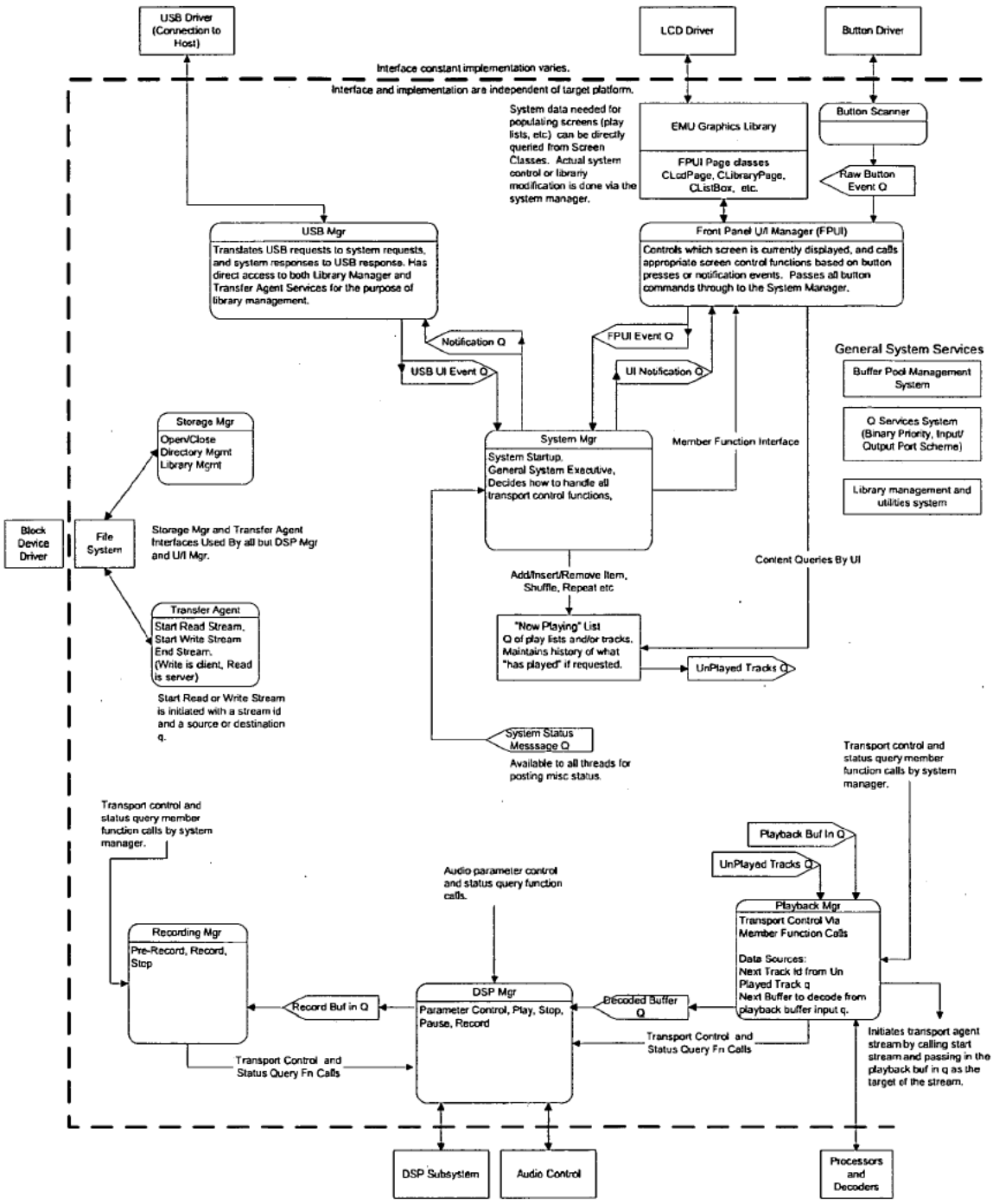
SIZE D | PC10239 | REV 1

MAT: NONE

# EXHIBIT S

# Oasis Operating System Software Diagram

Rev 5, 11/7/99

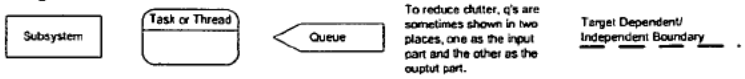



NOTES:

Two places that look light are the USB area and the Recording manager. Their exact functionality is not as well thought through.

Missing are: Power Mgmt, Boot Loader and Firmware Upgrade.

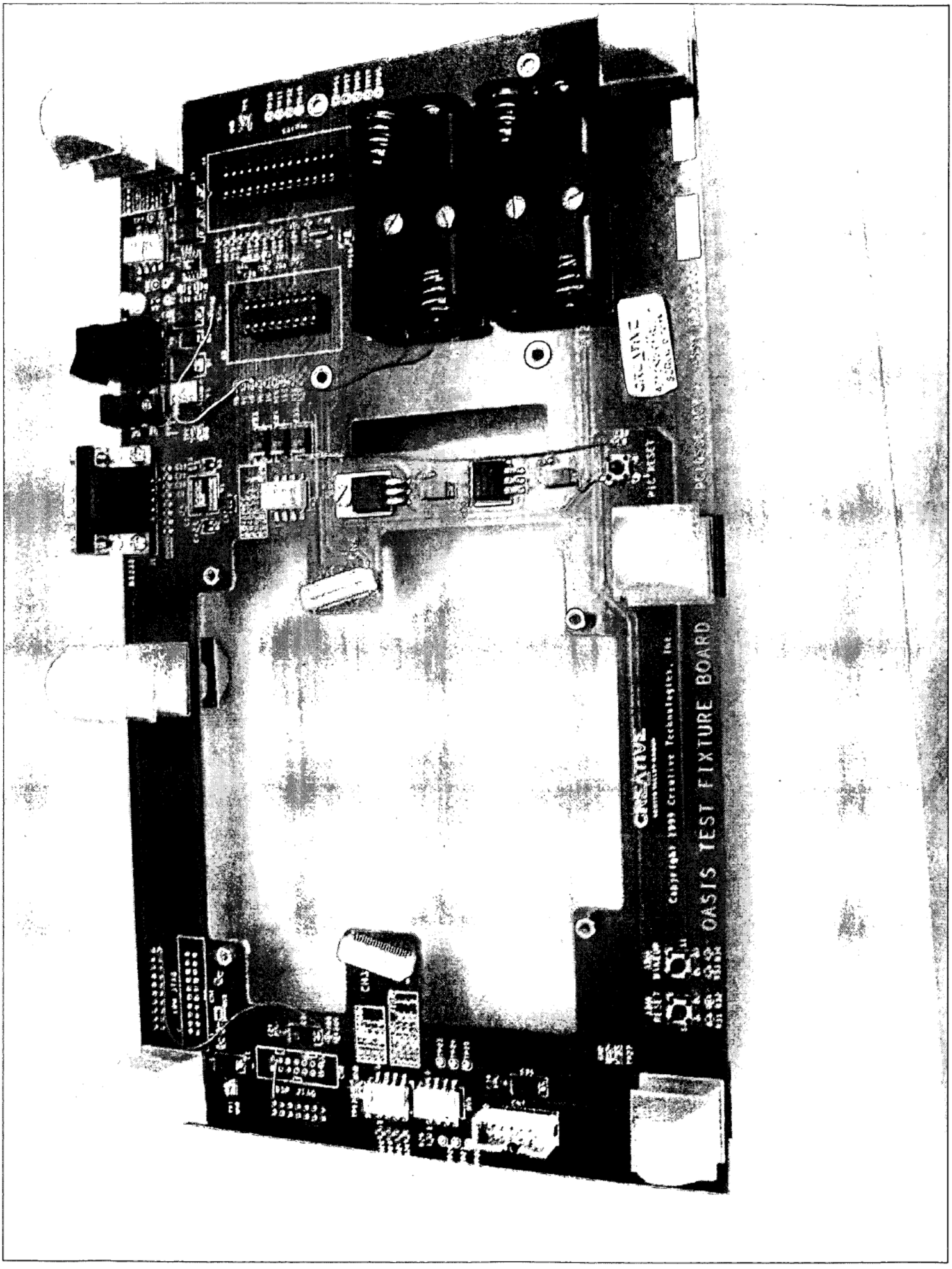
Legend

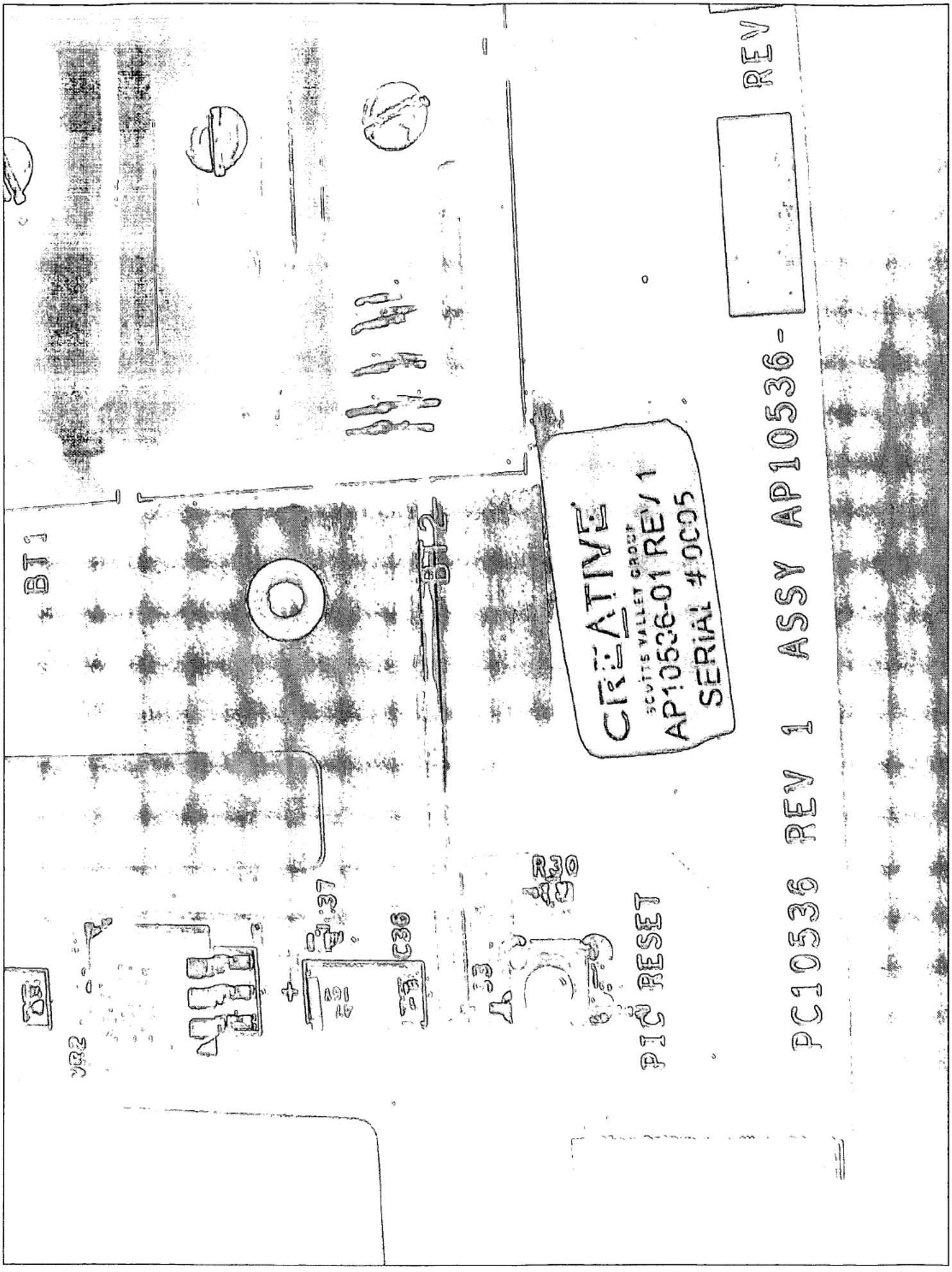


# EXHIBIT
# T

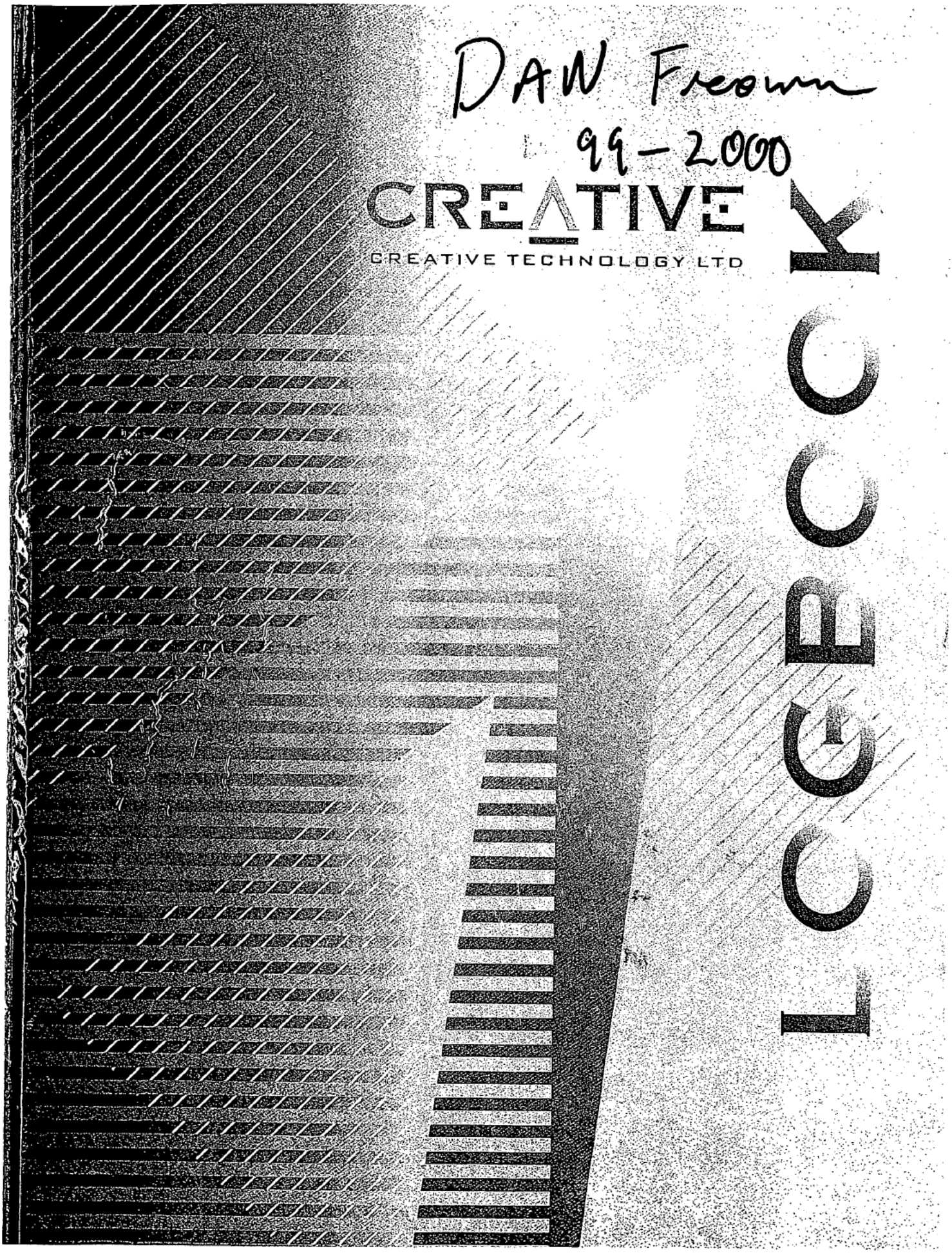
OASIS NOMENCLATURE
11-8-99 : THOMSON
LINE OUT
LINE IN
USB
KEEP THIS AREA OPEN
DC IN 12V
HOLD
(REAR SWITCH)
LIB
MENU

# EXHIBIT
# U





# EXHIBIT V


DAW Freeware
99-2000
CREATIVE
CREATIVE TECHNOLOGY LTD
LICENSE

Dan Freeman
E-mu Systems Inc.
1600 Green Hills Rd
Scotts Valley, CA 95066
(~~408~~) 430 1767
831
dan_freeman@emu.com
↑ underscore, not dash

CREATIVE

CREATIVE TECHNOLOGY LTD

Employee # 109344

Doo - Positronics Nancy Lucas 2+ Automail x5088
510 492 5088

name: Daniel Freeman
employee no: 109344
department: E-mu Engineering
date of issue: 12/3/97
logbook no: EE-109344-0007

NOTES:

The content of this logbook contains intellectual property of Creative Technology Ltd.

The pages of this logbook must not be removed.

Upon termination of service, this logbook must be returned to your head of department.

SUBJECT: DATE:

10/11/99

Syvex Meeting Agenda

① Power Supply prototyping - Syvex to fab prototype PCBA to prove out design?

② Thermistor issue - Syvex to take ownership of making power supply pass UL/FCC/CSA etc.

~~3~~ 3A Systems Review

③ Preliminary Schematic Review

④ Preliminary BOM Review - Dan F to order prototype parts for power supply subsystem

⑤ Schedule Review

PCB Layout of PwrSupply proto board:

- 3 weeks is soonest we could get board. Assumes parts are avail.
- Syvex has resources avail.
- Doesn't seem worth it to fab a whole new board just for power supply -

- Bill C could come on Thursday to help fix boot-up problem - RSVP ~~Wed~~, if no RSVP, Bill won't come over. Thursday 10AM ~~Thursday~~ 10/14/99 FRIDAY

☆ Lithium Batteries NOT ACCOMMODATED

CREATIVE
CREATIVE TECHNOLOGY LTD

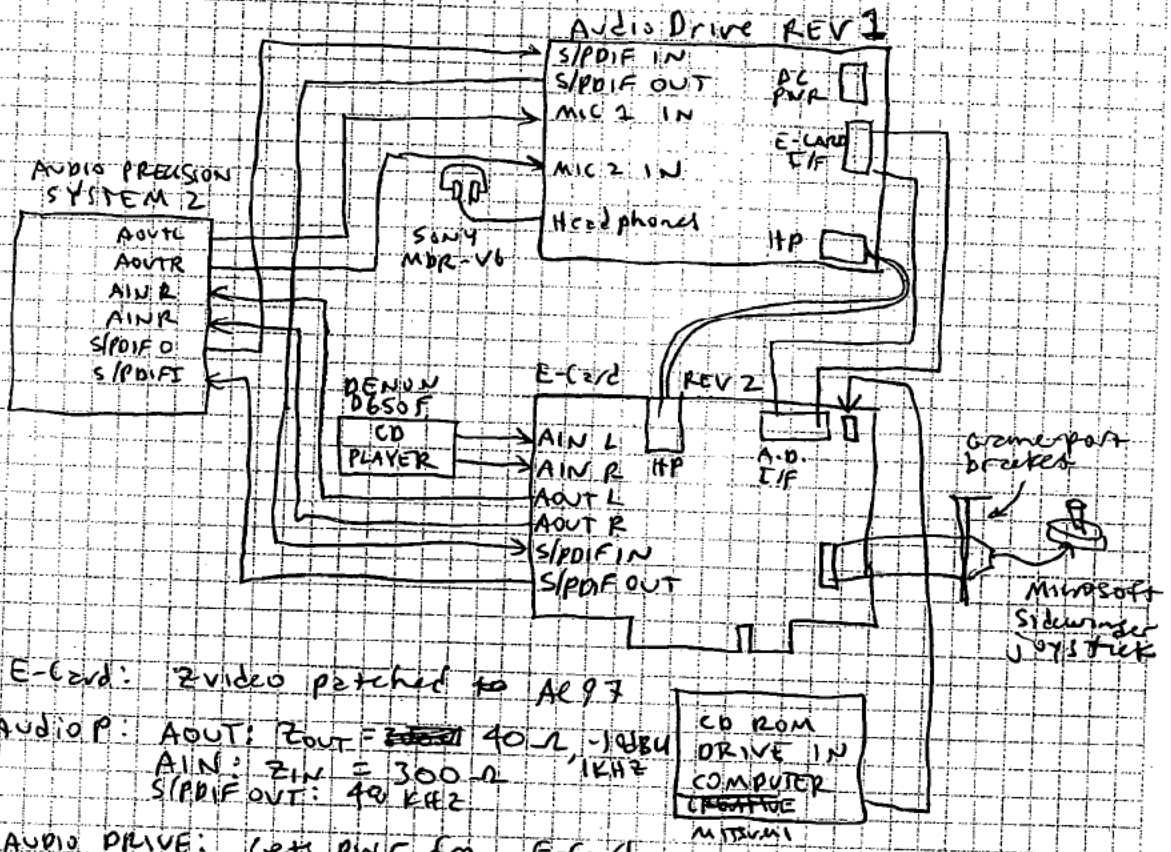
CONFIDENTIAL

2

SUBJECT: E-CARD DBF 12/20/97 DATE: 12/20/97

E-Card / Audio Drive Current Measurements

Setup:




E-Card: Zvideo patched to AL97

Audio P: AOUT: $Z_{OUT}$ = 40Ω, -10dBV, 1KHz
AIN: $Z_{IN}$ = 300Ω
S/PDIF OUT: 48 KHz

Audio Drive: Get PWR from E-Card
U3, D9, L1, ~~F1~~ NOT INSTALLED
FB9, FB10, FB11 are installed. Head phones full volume.

CD player: Both playing Audio CDs

Current Measurement Method: Use Adex electronics PCIX32 isolation extender card for PCI Bus. ~~Short~~ Replace fuses for +5, +12, -12V with solid conductor wires. Use TEK A6302 current probe, TEK Ams03 current probe amp, and TEK TDS350 Scope to take current measurements

SUBJECT: DSF 12/20/97 DATE: 12/20/97

RESULTS:

-12V: ~~300~~ 275 mA (Extender card fuse W4)

+12V: 450 mA (Extender card fuse W3)

+5V: 560 mA (current probe drifts like a mofo)

1125 mA

★ Current draw at -12V of 275 mA exceeds PCI spec of 100 mA.

RESULTS #2:

Use -12V inverter on Audio drive, disconnect -12V from E-Card interface cable.

→ APS46: Remove FB9. Install U3, L1, D4

BOGUS! -12V inverter circuit causes tons of noise on the headphones. Noise also induced on +5V.

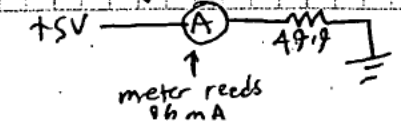
I don't believe the current probe measurements. Try Fluke 83 Multimeter instead. Calibrate meter

→ APS46: tie U3-3 to +5V to disable it. Install FB9.

-12V: 115 mA (This is close to PCI max over by 15 mA)

+12V: 280 mA

+5V: 532 mA



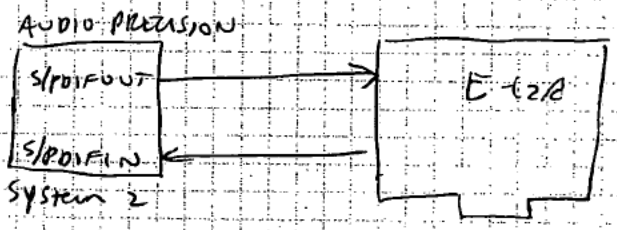

CREATIVE
CREATIVE TECHNOLOGY LTD

CONFIDENTIAL

SUBJECT: E-C2A- 8010 SRC LOCK BUG DATE: 12/23/97

# 8010 LOCK BUG NOTES

Setup




Audio P system 2: S/PDIF over BNC (RCA) unbal, 2.55 Vpp, 24 bits, 6' long Crappy audio RCA-RCA cable, -1 dBFS level.

E-C2A: GpSPDIF routed to SPDIFØ on 8010.
EMUX1 set to Ø in Control PAL

FAILURE MODES

8010 sample rate tracker loses lock when S/PDIF input is close to 48 KHz sample rate.

Can watch SRT behavior by polling 8010 register Øx61, as follows:

GD8010> wr hc 18083
> outw 7ØØ2 61 FO BASE: FCCO
> Poll fg 7ØØ4 4 none

Port: 7004 Data: 0303ffce
↑ LOCK bit
↳ frequency
40000 = 48 KHz

(from the GD16api.exe program)

SUBJECT: 8010 Lock bug DATE: 12/23/97

The 8010 SRT loses lock at the following frequencies:

47990 < S/PDIF SAMPLE RATE < 48010 Hz.

At 48 KHz, SRT loses lock every 5 seconds or so.

---

1/3/98

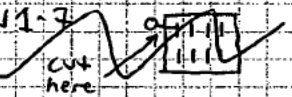
E-Card CD ROM S/PDIF Input Fix-

Rework Instructions & Test Results

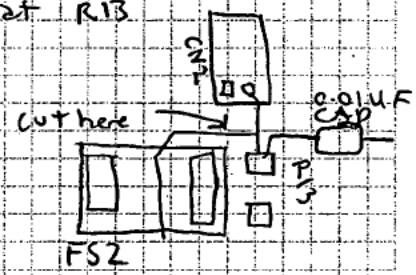
Rework: (See schematic SK545 Rev 2 SH1 & SH2)

✓ ① ~~Remove R33~~ Install 75Ω Resistor at R13

✓ ② Cut trace from RN1-7 to CN2-8 at RN1-7




✓ ② CUT trace from CN2-2 to U11-20 at R13




✓ ③ Remove R73. Install 330K resistor at R73

✓ ④ connect 22K resistor from U16-13 to U16-12

✓ ⑤ connect U13-12 to U11-20

✓ ⑥ Attach 0.01 uF cap to R13/CN2 connection

✓ ⑦ Connect other side of 0.01 uF cap to U16-13

Test Results

Connect S/PDIF output from MITSUMI CD ROM Drive

SUBJECT: E-Card CD SPDIF Input FIX DATE: 1/3/98

(Cont'd)

Model # CRMC-FX1405Z to E-Card CDROM S/PDIF input using twisted-pair 2-pin cable. WORKS. Without above rework, E-Card CD S/PDIF input will only work if COAX cable is used.

(Above fix also works with COAX cable)

---

1/6/98

A1 8010 silicon - Needs new GD16.ERB program.

See c:\projects\gd8010A1\gd16\emi

c:\projects\gd80102A1\

---

1/12/98

E-Card meeting

✓ (1) Draw Diagram for John Kraft GD16 Training Session

✓ (2) Call John K about ↑

(3) Get ready for Design Review, MON 1/19/98 (FRIDAY 1/16?)

(4) Schedule Design Review for ↑

✓ (5) Write rework instructions for APSAS S/PDIF FIX

✓ (6) Order bench.

SUBJECT: DATE:

(cont)

⑦ Update HW spec

✓⑧ Manage APS45, APS46 bringup

✓⑨ Meet w/ chuck & Steve S. about FCC testing. → Brent E. will take action item

1/16/98

Audio Precision CAR # is 986801-06

Audio Precision P.O. -

Buy Tone burst generator option instead of IMD analyzer.

Audio P 18500
AWT
CCIR
Brickwall
SIA → change to S2-BUR tone burst gen.
HANDLE

Delivery Date 2/3/98 Mike Hogue 475 6291
Can keep Loaner until then.

SUBJECT: DATE: 11/19/98

APS45 Rev 2 S/PDIF Fix Rework Instr.

Bug: Audio Drive gets its S/PDIF from 8010 Chip SPDIF0, Should be SPDIF 1.

Fix:

① Lift U3 pin 2

② Jump U3 pin 13 to lifted pin U3-2
Remember 300K R Fix to S/PDIF IN!

---

1/19/98 E-Card HW Meeting

My Action Items:

① Verify that I have an Ecard w/ A0 8010, tell Jeannie which S/N.

② Get training from John K on 8010 Drivers. Make sure support is included to support continued HW verification testing.

✓ ③ Write email to Mike P outlining requirements for PC computers for Ecard testing. (Gave Mike P-Req thumbs)

④ HW Spec Monday 1/26/98.

⑤ Schedule with John K. to give 8010 SW training session - Thursday 1/22/98?

---

→ ③ Install 300K at R75, not 330K

SUBJECT: DATE: 1/19/98

Audio Drive APS46 Rev 1 Rework:

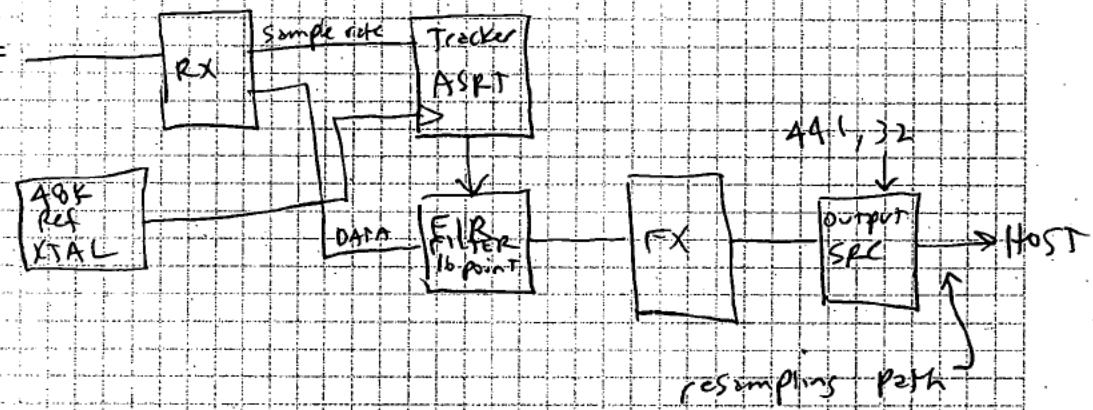
① Install 300K at RS, NOT 330K

---

1/20/98

8010 chip - Clocking issues.

Can clock 8010 with a 44.1K clock, but MIDI won't work, and the resampling path won't work.




---

E-Card Eng Meeting 1/20/98

Brent, Aime, Me

My Action Items

① use MMBZ6V2ALT1 for static-zap protection.
Buy parts Change BOM → Don't change BOM.
parts don't work as BIDIR protection because
diodes in MMBZ5V6 are reversed from MMBZ15.

NO BOM'S!

CREATIVE
CREATIVE TECHNOLOGY LTD

CONFIDENTIAL
ACTION REQ'D

10

SUBJECT: DATE: 1/21/98

DMV 476 7480

Monday Jan 26 8:20 AM

CAPITOLA DMV

CONF # WSSØØ126Ø92Ø

Licence ID Card Appt. window

---

E-Card S/PDIF Performance Panic Meeting 1/21/98

My action items:

✓ ① Estimate cost/schedule of adding SRC chip to Audio Drive.

↓ $6.00-$10 per board, 1 mor month

Due Monday 1/26/98, give results to Brent E.

Idea: route Internal CD ROM S/PDIF to Audio drive so can SRC it.

---

Analog Devices - Linda M 1/21/98

AD1892 - will give quote & lead times QTY 500/Mo, max 2500/Mo. Via email.

Ken, Hank Z. are contacts. Never'd

EVAL board or way.

---

CREATIVE
CREATIVE TECHNOLOGY LTD

CONFIDENTIAL

SUBJECT: DATE:

1/22/98

Scott Fuller - 8010 specs: All signal paths are 20 bit except SRC's and the sound engine.

---

SAMO (650) 655 4126

Music & Computers Magazine

Wants "Visual" for writeup on Audio Production Studio in Magazine → Bill S, Dennis L will provide data. 1/22/98

---

E-Cards SW Meeting 1/22/98

1. Advertise how Card Detect Mechanisms Work in Technical Spec and in SW Programmers Guide.
2. Prioritize changes to HW that will require support from Control PAL. Should we implement EEPROM programming support?
3. ✓ Send email outlining new support req'd from gd16.exe program. ADC Input routing
4. ✓ No training session on 8010 drivers avail. until beta SW Drivers available, approx 2/20/98

---

SUBJECT: ECard DATE: 1/29/98

# E-Card AC3 Input experiment

Goal: Figure out how AC3 inputs on 8010 work so can debug Digital Audio Interface header works on E-Card

Experiment:

Connect ZVIDEO data to AC3SD1 input, Program 8010 to replace the CDSPDIF data with data on AC3SD1.

① Jump CN1-13 to CN4-13. (ZVIDEO serial data to AC3SD1)

② Program the 8010 HC register to replace CDSPDIF with AC3 channel 1, enable phase tracking of the three SRC's.

in gd16.exe:

```
wr HC 10003
outdw 7014 00010143
epatch CD 2c97
```

The CD SRT appears to not lock to the ZVIDEO SRT. Get garbage audio.

③ Jump CN1-13 to CN4-15 (ZVIDEO DATA to AC3SD0).

④ OUTdw 7014 00010123 (Replace ZVIDEO with AC3)

→ still doesn't work. SRT doesn't lock?

SUBJECT: DATE:

1/29/98 8010 SW Meeting

→ Discuss HW Drivers for APS45 Rev 3, Ed's SQA efforts.

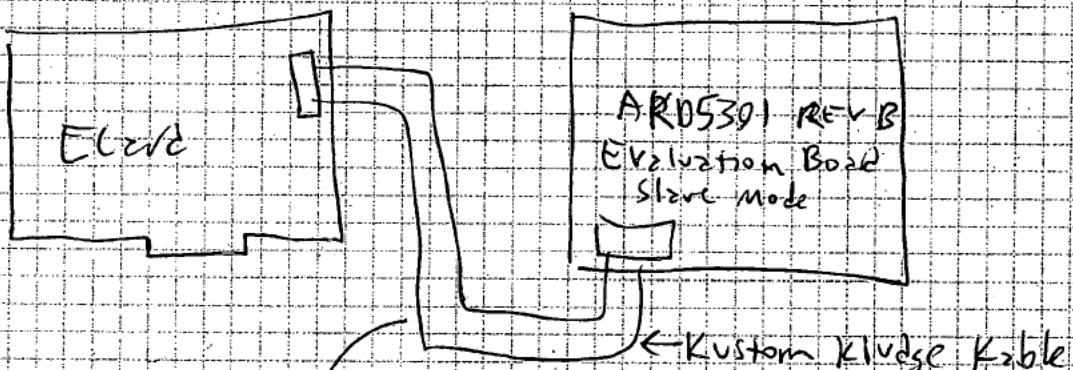
MY ACTION ITEMS

① Design in ability to program EEPROM in-system, if possible.

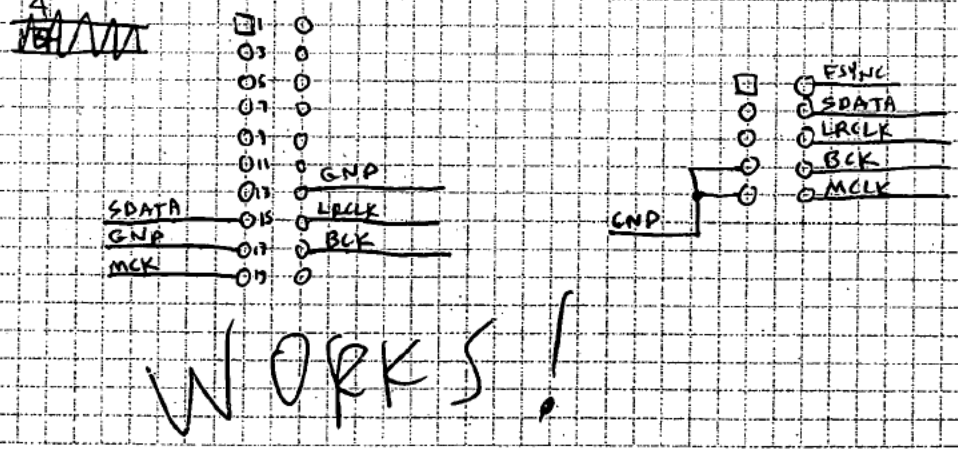
---

1/30/98

Eczid Dig Audio Exp I/F test Setup




Cable pinout:




WORKS!

SUBJECT: DATE:

2/4/98

Gave John Kraft #16 eCard w/EEPROM programming rework

---

2/5/98

E-Card Design Review

Action Items

✓ (1) Gnd unused S-VIDEO Inputs (DAN)

✓ DON'T NUKE (2) Revisit Nvkins S/PDIF XFMR

✓ (3) Allow bypass of Joystick comparators.

✓ (4) ECard HW spec: PCI Specific 8010 is Universal PCI (5V or 3.3V) UNIVERSAL

✓ (5) Next Rev Card - Make Universal Card?
Must hook PCIV10 from PCI Con to PCIV10 On 8010, add another wrong on PCI edge connector. (DAN - Read PCI Spec)

✓ (6) Review which pins are 5V tolerant on 8010 DAN ~ Ask Brian H

✓ (7) Provide data on testpoints - BRENT

✓ (8) Short AGND to DGND at FB36

✓ (9) Implement 3-wire interface to DIG EXP I/F

SUBJECT: DATE:

Running Allegro remote on Exceed 2/11/98

pcs45/rev3/pcb/ source ~dang/.cshrc
/source ~dang/cshrc12
/pcb PCS45_3

Pw: RMB

2/12/98

HP VECTRA ALLOCATION

#1 ELAB

#2 ELAB

#3 ED SQA

#4 JOHN MIRRASOU (SQA #2)

#5 ELAB

SUBJECT: DATE: 2/17/98

## HP Vectra Computer Setup Checklist

① Install CD ROM in TOP Drive Bay

② Install Network Card in BOTTOM PCI Slot

③ Install Sony Monitor Driver so can change display settings

④ Install Netscape so can access E-mu ~~Web~~ Web Sites for ECard technical Documentation

---

3PM TEQ Tom Hendricks - ECard Prod. Test Mtg.

---

ECard Meeting 2/17/98

We will fab QTY 100 PC545 Rev 3.

Duane Ford - Ask about ECard Web.

---

2/20/98

E-mu ECard Web page: emuweb.emu.com/WebServer/index.html

SUBJECT: DATE:

2/10/98

# ECR APS45-02 BOM Notes

Changes from APS45-01 Rev 2 BOM

| | | |
|---|---|---|
| RP409 | 4.99K: | √Add R1, R2, R8, R9, R10 R14, R15, R16, R18, R19 √Delete R42, R52, R54, R62 |
| RP402 | 1.0 K: | √Add R1000, R1001 |
| RP392 | 10.0 K: | √Delete all |
| RP400 | 33.2K: | √Delete all |
| RP424 | 2.26K: | √Delete all |
| RP426 | 715Ω K: | √Delete R1, R18 |
| RP421 | 39.2: | √Delete R10, R16, R44, R39, R40, R41 |
| RP433 | 150.0: | √Add R39, R40, R41, R44 |
| CC416 | 3900P: | √Delete All |
| CC392 | 2200P: | √Delete C36, CA3, C52, C66 √Add C38, C39 |
| RP434 | 13.7K: | √Add R42, R43, R52, R53, R57, R58, R60, R62 |
| RP432 | 3.32K: | √Add R45, R51, R56, R59 |
| CC415 | 680P: | √Delete all |
| CC430 | 220P: | √Add C37, C40, C62, C65 |
| CC391 | 1000P: | √Add C36, CA3, C52, C66 |
| DD333 | BAT54S: | √Delete all |

SUBJECT:

DATE: 3/28/98

Ensoniq Meeting

Ensoniq Personnel:

Carl Bader
Dan Genest - MKTG?
Bill Mauchley - Engr?
Dave Nutting
Scott Beer

Co-engineering opportunities discussed:

(1) Implement I/O card for IVY that has SCSI (cable) I/F to EDS 1000 PCI card.

(2) EDI card for ~~Mach~~ MEC.

SUBJECT: DATE: 5/8/98

Mike P Meeting.

Agenda:

- Personnel issues - reports, time cards, salaries, etc.
- Ivy Staffing
- Dept Budgets - How to work up?
- Ivy control surf. disconnects
- Status reports to Mike?

Kristy - Send me $ & LO Dept.
Negotiate w/ Brent On Mark C.
Consultant House for Option Cards.

{ Jim/Ron transition to PARIS integration?
Turnkey PC + PARIS + IVY for $10K
based on Creative PC }

5/13/98

Mike's Staff meeting

- Need budget info ASAP
- Budgets - Bottoms-up Eng budget is 6 million. Only 6.1 M available, but need to fund new business + Sounds. So → Kill or delay one or more projects. Delay Steinway, Kill e-synth KB, or Phatt product.
- ECard is "imploding". Another 10K1 bug? A4 version of chip is required.
- R&D budget too high. Need to recalendarize expenses, more them out of 1st quarter.
- Staffing: Want intern for eng., HW eng (new hire), financial Analyst, Doc Control person.
- Operations Review in June. 4TH week in June? NOT last week in June.
- Managers Meeting - Was it today? Guess not.

CREATIVE
CREATIVE TECHNOLOGY LTD

CONFIDENTIAL

SUBJECT: DATE:

Mikes Staff Meeting (cont) 5/13/96

- EOS 3.0 bugs - Stop-ship in effect. Lots of talks but seems to boil down to 2 bugs - Playing external sequence from ROM instead of RAM, & 3.0 crashes "more often than 2.5".
- E-IX bugs - Hot systems crash with SGS 68901 MFP chip.

---

Ivy Engs Meeting

Meeting time change conflicts

Craig - Wed 10 AM

Ron - Tue Thu

→ late pm desired

- DANO cntlr? E4 1/2 FPGA work. Will implement Interrupt
- Graphit - ADAT FPGA work. Outputs work. Implementing channel assignment scheme. Targeting to 10K30 FPGA ($33). Picked up control surface - sized costing, talked to Ken Yippiela.
- Ron - lots of progress on Dynamic Automation. Used Ivy to mix soundtrack for Cabrillo College version
- Jim - MAC O.S. download works. Jim's O.S. got corrupted during download. Will make O.S. Checker more robust. Add reset pushbutton?
- Craig - P-Chip Emulation FPGA work to exercise P-Chip. Wants more time to write VHDL test modules.
- Tom - Week off. Moved office back to E-mu. late May

SUBJECT: DATE:

# Ivy Budget Worksheet

5/14/98

| PCBA's | Electronics Cost | (Steve T. EST) Mech Cost | Tooling |
|---|---|---|---|
| MMM | | 400 | 2K |
| MAIN | (500)(1.5) = 750 | | |
| ANALOG | (100)(1.5) = 150 | | |
| PWR SUPP/CABLES | (50)(1.5) = 75 / 975 | | |
| ANALOG | | 250 | 4K |
| MAIN + EDI | (250+150)(1.5) = 525 | | |
| PWR SUP/CABLES | (50)(1.5) = 75 / 600 | | |
| ADAT | | 25 | 0.5K |
| MAIN | (175)(1.5) = 263 | | |
| TDIF | | 25 | 0.5K |
| MAIN | (175)(1.5) = 263 | | |
| EDI | | 25 | 0.5K |
| MAIN | (150)(1.5) = 225 | | |
| SMPTE | | 25 | 0.5K |
| MAIN | (1500)(1.5) = 225 | | |
| 7 PCB's | $1675 | 750 | $15,000 |

## CONTROL SURF WORKSHEET

| PCBA's | Electronics Cost | (Steve T. EST) Mech Cost | Tooling |
|---|---|---|---|
| Computer | | 700 | 40K |
| PANEL #1 | | | |
| PANEL #2 | | | |
| ENCODER | | | |
| METERS | | | |
| PWR SUPP/CABLING | | | |

SUBJECT: DATE:

5/19/98

DO THIS -

Try to get VHDL models for Sync SRAMs:

① Micron MT58LC128K32B4

② Cypress CY7C1351

---

5/20/98

Ivy Lunch Mtg

Circuit • Cont. Surf cost rollups. Current list is $3400 - 4 kes.
• EDF UART prototyping

Tom S - • Ivy Rack - VHDL design of sync subsystem. • Back at E-mu to work instead of home.

Ron
• Scene Automation development. File system heirarchy w/ Jim. for Projects, libraries, etc.

→ { out back and kerten. Jean LaRoche, for ditching out of Bike-to-work Day.

Jim
MAC bug fixes. Graphics impact.

Craig Tech center guys are back on R-Chip. New emulation effort starts June 2. First date for 1st silicon is 10/14/98. Craig working on FPGA on emulation board. Need to implement S/PDIF input, will use S/PDIF I/O VHDL module from Tech Center.

SUBJECT: DATE:

Craig (cont) New pin on R-Chip! Get from Craig. Needs help w/ synchronous SRAM support - get samples (dsm), make adapter board so can prototype. Will work on ASIC release process for E-mu VLSI

Dsm

DI stuff. Dept Budgets, Int Cntlr studies, etc.

Balance Sheet

\+ —

Good food.
best choice

Finished on time

Someone to take out & beat

Someone likes balance sheets

Someone doesn't like balance sheets

Dsm O leaving

SUBJECT: DATE:

1U4 MMM Mechanical Design 5/20/98
Kickoff meeting.

Dan, Steve T, Grault, Tom S.

Steve took input on 3U chassis design features.
7 slots, "roadworthy" option, etc.

Dan corrected McMike's MRD V1.0 to Rob's concept drawing of MMM Chassis 2/15/98. Only difference is LED's to - option slots 6 & 7, EXT. lock.

5/20/98

Order: Cypress CY7C1351-40AC
Micron MT58LC128K32B4-10LG
Aries 96-100M65 QFP to PGA Adapter (Digi-key A175-ND)
Mil-Max 515-13-121-13-061 121-pin PGA Socket
(DIGI-Key P/N ED2027-ND $5.61)
~~Aries Electronics 96-100M65~~

5/27/98

Mikes Staff Meeting

- Recruiting - Nikl made offers to 1 full-time & 1 part-time Summer Intern for SW.
- → Dept Meetings Thursday 3:30
- Monthly Income Sucks - Dan Gerrit wants to double the number of devices.
- Budgets are being finalized → SW Development Head
- → Call Scott Reed @ Ensoniq about 1U/PARTS interface Carl is HW engineer
- List of vacation plans to Mike P.

SUBJECT: DATE:

- E-mv/Echo joint venture in Italy. Ensoniq "Bumble Bee" Digital Piano might be mfg'd there.

---

Ivy Lunch Meeting 5/27/98

**Mr Mike**
- Lots of interest in Ivy at AES in Amsterdam
- Tons of problems with PC. Ron's program crashed.
- Mackie wants us to OEM HUI, but wouldn't let us show it at AES.

**Jim W**
- Fighting gnarly crash bug.
- How soon will automation be available on MAC version of SW? Long time. Should port automation feature into Rack SW
- Gone Friday & 1/2 next week.

**Ron C**
- File ~~hierarchy~~ Hierarchy, Libraries, projects, etc.

**Circuit**
- Control Surface - Cost rollups & System Design.
- EDI VART stuff - 95% done?
- Cont Surf cost is about $3500 list now.

**Tom S**
- Learning VHDL design tools
- Clock/sync system implemented in VHDL
- Vacation Friday → all next week

**Dan OV**
- E 4/2
- VHDL - Learning tools

**Craig**
- Liar
- Met Mike P. in bathroom.
- Tech Center NOT working on R-Chip. Fixing 8010 bugs.
- about finished w/ emulation FPGA.

SUBJECT: DATE:

Mike P Budget Meeting — K. Long Creative CFO

Budget Requirements:

- Quake over quake improvements
- Q4 rev-rese + 1 million
- Make $500K PBT

---

Mike P. Ivy meeting — Dan F, Steve T.

- Investigate ESPII Ensoniq chip? DSP chip used as FX engine. Core DSP engine in PARIS
- Cost - $7 Schmitt Feder??

Control Surf:

Girault/Steve T — Spec, costing
Scott R 7/1/99: Proj Mgr Control Surf
Ken Y: Implement
Summer student: Electronic implementation

MMM:

---

Ivy Lunch Meeting — 6/3/99

Rob

- Publish SW schedule + milestones?

Dano

- E4.5
- Flying Grandma

SUBJECT: DATE:

Ron

- Auto match mode
- Project Hierarchy
- Feature additions based on feedback from AES show

Craig

- 8010 final signoff Friday?
- R-Chip work resumes Monday?
- 1st silicon 10/19/98 ?
- Updating R-Chip Spec for synchronous SRAM.
- Emulation HW done.

Grault

- EDI
- Control surface cost estimates w/ Steve Thompson
- Latest cost based on ML Mike concept is $3081
  - No LEDs inside switches
  - 2 ch LED scribble strips
  - $11.50 Faders
  - $1.50 encoders
- Feature-axed version is about $2500.
  - No LED rings - bar graphs instead
  - CS16 style graphics

---

6/4/98

Grault: Ivy Current UML's

| | UML | contingency built-in | |
|---|---|---|---|
| MMM | $1020.77 | $73 | (Includes TDIF card) |
| ADAT | $165 material | | |
| TDIF | $120 material | | |
| Analog I/O | $558 | $48 | |

SUBJECT: DATE:

6/5/98

Mike P, David B, Mc Mike, Dan F Ivy meeting.

- R-Chip schedule poses unacceptible risk to Ivy ship date
- Need to look at ESP2 spec (Ensoniq DSP) & see if we can design it in instead of R-Chip.
- Last possibility is to "jumper over" R-Chip
  - Alan / Scott Peer / Carl Baker - Contact about ESP2 chip.

---

6/9/98 IVY - ESP2 vs. R-Chip Discussion. Mc Mike Circuit Craig A Rob B Dan F

Use ESP2 FX chip by Ensoniq instead of R-Chip to mitigate project risks imposed by R-Chip schedule

Assumptions:

R-Chip System on MMM:

| | |
|---|---|
| 1 R-Chip | $15 (amortized) |
| 4 128K x 8 SRAMs | $20 |
| | $35 |

ESP2 system on MMM:

| | |
|---|---|
| 2 ESP2's | $26 |
| 6 128K x 8 SRAMs | $30 → Is this correct? |
| ~~1~~ 2 FPGA's (10K30) | $66 |
| | $122 |

Raw parts cost Δ is $87 if use ESP2
↳ does not include Δ to TDIF card in MMM.

SUBJECT: DATE: 6/[illegible]/98

## ESP2 PROS

- ESP2 EXISTS
- FX Algorithms "Done"
- Alleged plug-in architecture
- Faster time to MKT for ESP2 in MMM?

## ESP2 CONS

- 1/4 the HP of R-Chip
- Cost prohibitive
- SW complexity of supporting code - Schedule prohibitive
- 50% feature hit on FX capabilities
- 24 bit vs. 32 bit precision on DSP Math
- Not much in-house expertise on developing FX algorithms on ESP2
- FX algorithms harder to write.
- HW design schedule risk

## R-Chip Pros

- Ivy architecture based on R-Chip Design - R-Chip is Ideal Solution.
- Most Cost-effective solution
- 32-bit processing
- Meets Ivy Marketing requirements

## R-Chip Cons

- Ivy architecture based on R-Chip design
- R-Chip Schedule: Current 1st silicon 10/19/98
- R-Chip might not work the first time.

SUBJECT: 6/19/98 DATE:

CONT

The "Plan"

① Maintain Product architecture w/ R-chip

② Accelerate R-chip Design-in:
- ⓐ Pressure Tech Center
- ⓑ Reduce R-chip DUT from 2 months to 1 month (RISK)
- ⓒ Reduce Beta from 2 months to 1 month (RISK) ⓓ reduce # of FCS FX

③ Ship Ivy without Rchips (Contingency Plan)
- ⓐ R-chips optional stuff on Option cards
- ⓑ R-chip in MMM on internal option card. Card can have outchip instead of R-chip (UMC hit)
  - ↳ to mitigate R-chip schedule risk

---

Brent E: 6/19/98

J. Schwitalla - Consultant guy who could design Analog I/O module. Dir of Eng at Diamond Audio.

---

Mike P. Staff Meeting 6/19/98

- Summer students - think of assignments for them. High impact projects only
- New budget sent to Creative. 41M sales, 45% margins, expenses at Q4 run rate, 1M product launch expenses 8M sales for new products.

SUBJECT: DATE:

- $6M budget for R&D vs 4.7m FY98
  450K sound dev, 300K consultant exp, etc.
- Cash crunch. May may break even, April lost $300K
- Cash recovery out of inventory is important.
- Next products:

| | |
|---|---|
| APS | July |
| APS "Lite" | September |
| E4.5 | October |
| composer | Nov |
| Moby II | Nov |
| Ivy | Apr |
| Sessions | Apr + |
| Orca | June |
| steinway | June |

Sound Module EPanII.simm

---

Ivy lunch Meeting 6/10/98

- New meeting time: Wednesdays at 3 PM

6/10/98

- Ron - New features for HEA based on MI Miles feature requests
- Time out to document code.

JM

- working on schedule.
- got HEA working on MAC via virtual PC

SUBJECT: DATE:

Ivy Meeting (cont't) 6/10/98

CRAIG

- Risk wafers for R-Chip - might be able to get R-Chip 1ST silicon by 10/1/98

Calvin

- Control Surface Cost estimates completed, $3100 model meets MKtg reqmnts, $2500 model has feature cuts. "New" cost model used
- EDI work.

---

Mike's Staff meeting 6/17/98

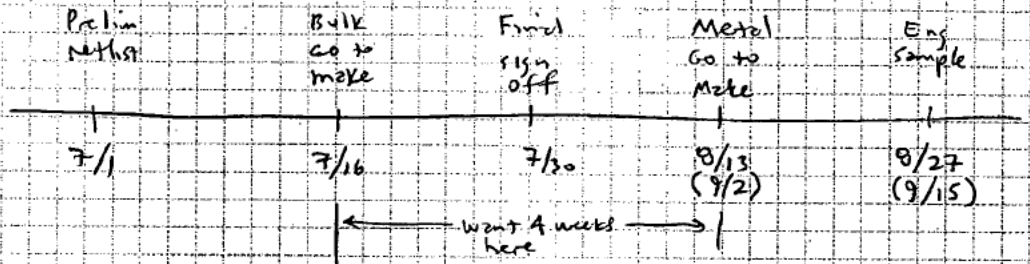
- New budget to K. Long at Creative, he didn't object
- Lose $1M in 1ST half make $2M in 2nd half Profits depend on Ivy and EA.SXP
- Trying to flush out obsolete inventory to get cash.
- Product mix fluctuations - Steinway bumped.
- watch out for designing in obsolete p2As! 82C80, 82078, 256K x 4 DRAM.
- EA.S Project understrain - schedule pushing October
- Composer next
- Nil - EA.S design - turn into keyboard. Should be no SW port req'd
- OPS Review - Looming
  Presentations: About 10 min, including Q&A. Say 15 min. This means eng presentation time is about 5 min.
- Oracle Demo

SUBJECT: DATE:

# R-Chip Meeting w/Fujitsu

Brian S., Duke Powers, Kory Khosravi, Craig, Dan F, Carl W, Vincent Shen, Scott F, Daniel M, Jake, Barry

## Prelim New Schedule:




need to know # of wafers.

↓

Place P.O. for
- # of ES
- # of AES
- # of CS/prod

} several hundred chips Available

- Scott F sez we're about 80K yields, (no Silan insertion.)

---

Ivy Eng Meeting

6/22/98

Jeznan

- Going Ivy parts - lots of stock!
- Is Tom ordering parts? Heads up!

Tom

- Always Late to meetings

SUBJECT: DATE:

## Mc Mike

- Showed Ozmo in LA
- Set up Ivy in posh Studio
- Poor attendance, Sold maybe 4 demos.
- Got HCA Suggestions. No HCA Crashes

→ We need a MRD for the Analog I/O module

## Tom

- Push off parts ordering until layout
- Next week - try for white paper Prelim BOM
- Design Rev Monday 6/29
- Computer System design is gnarly.
- Need 5307 ICE

## Circuit

- delivered EDI card to Rob so he can write drivers.
- Resolved AC coupling problem
- Got system ready for FCC testing, Chuck to work on it.
- Got Cold
- Sample secrete time-stamp brainstorm w/ Tom & Scott B

## Craig

- Good schedule news: Eng Sample 9/15 best case
- Emulation HW in Tech Center ~ ready for emulation.
- Fujitsu got prelim netlist

[ Fu Chip volumes: FX processor, E5 ]

## Rob

- New concept drug for MMA (slick!)
- Clock subsystem
- EDI card support
- Make Rory life Hell

SUBJECT: DATE:

Ron

- Last panel objects now platform ind
- Synchronization w/version of Rob's OS.
- Documenting code
-

Jim W

- FIX MAC bugs in channel strip
- Linked channels now work
- User-defined channel layouts
- Investigating Bzwk rate Problem
- Library functions.
- Debug window fix.

---

Mikes Staff Meeting 6/24/98

Mike, Penny, Dan F, Steve S, Rob B

Agenda

Roles
Product map
Op Plan progress reports
Development steps
  Composer
  Ultra
  Ivy
  Ozwin
New Business
WV + BS

- New Product map + forecast passed out. Sound modules biggest chunk.
- Plan includes Ivy Analog line in module, which we do not intend to implement. Disconnect.
- Map will be extended into FY 00 next qtr. We'll come up with budgets & schedules then. Each QTR we'll extend product map & business plan by 2 QTR.

SUBJECT: DATE:

(Cont)

- Thunder? Feasibility Study continuing, low priority. New Drum head prototype in process. Project is not committed. Technology Investigation Only.

## OP PLAN REVIEW

- Next Staff Mtg - review progress on presentations.
- New Ops Review Agenda will be published next week.
- Extended by 2½ days to allow time for Q&A.
- Some project will go first, to set stage for level of interaction desired, for "Role Model".
- Ivy will probably go first.
- Mike will present Ens Ops Plan

## Development Status

Vitrz: ~~Tweeter~~ AKA E4.5 ~~Project~~ AKA Rocky II

- Unhappy surprises on PCBA. Obsolete parts.
- PCBA out for FAB & Assy
- Tony in Vacation for 2 weeks
- Dan's FPGA work going well.
- SW: Brian Ahead on SCSI & Smidi. SCSI sample dump is 6X faster!
- Colin reversed AKAI bank format
- SW Port to Coldfire difficult. Lots of code restructuring req'd, unanticipated. Machine Dependent code.
- Device driver work req'd for new Coldfire implementation
- Mechanical design is done.
- Need to jump on FCC Scanning early July.
- Hire summer student to work on ADAT ~~I/O~~ output card for Vitrz. Andy.

SUBJECT: DATE:

Ivy

- Tom S. underway on new Coldfire design & mmm PCBA design. Behind sched
- Steve T has new mechanical design - 7 option slots.
- Getting Pkg together for Ken Yppariila for Control Surf design.
- J Schwiftlitz Consultant Signed NDA for working on analog I/O module
- EDI Prototype working, Chuck P

DAN - Schedule for PCBA designs (ME) for Denny G. } Vacation submissions

- Need to get Tom H involved in Control Surf

Darwin

- SMPTE SW release hopefully out next ~~week~~ month. Included Auto punch feature.

New Business

- APS - Manual needs to be re-done, bug fixes, FCS right at the end of July?
- APS forecast is 525. will issue P.O. to Solectron for Min Qty of 2000.

Other Items

- Dept Meeting Thu 6/25/98
- Roy will talk at Dept. Meeting.
- Moby not selling through in Germany? Bad review, too expensive?
- long discussion about new component designs. RESIST designing in new stuff.

WW

- MFG people only show up to this Mtg on 1st Monday of each month for SAS meeting.
- Program Management in seperate meeting - Penny will start

SUBJECT: DATE:

(cont)

Balance Sheet

\+ | —

Agenda published & followed.

low attendance - Vacations. Brent AWOL

6/24/98 Vacations

| Vacation | Dates | PAN Dates |
|---|---|---|
| Jim W | 4 days end of May + 7/7-7/10 + 2 days August | 7/4 - 7/16 (8 days) 8/2 - 8/? |
| Don F | 7/6 - 8/3 | PAN Submitted |
| Ron G | 10 days vacation, 10 days unpaid } 1st QTR | PAN submitted |
| Tom S | 7 already, 3 more Q1 | |
| Walt J | 10 days vac 1st QTR | PAN needed |

CREATIVE
CREATIVE TECHNOLOGY LTD

CONFIDENTIAL

SUBJECT: DATE:

| | | |
|---|---|---|
| Jim W | 10 days Q1 | PAN Submitted |
| Dan F | 20 days Q1 | PAN SUBMITTED 8/6 - 8/3 |
| Ron G | 10 days vaca<br>10 days unpaid } Q1 | PAN submitted |
| Tom S | 10 days Q1 | PAN Submitted |
| Circuit J | 10 days Q1 | PAN SUBMITTED 7/14 - 7/17<br>8/24 - 8/31 |

Daniel McDermott 6/24/98

R-Chip Buy Schedule:

| | |
|---|---|
| QTY 5 | 9/15/98 $87K (NRE) |
| QTY 25 | 9/22/98 $50/EA |
| QTY 75 | 10/5/98 $25/EA |
| QTY 150 | 11/15/98 $??/EA |
| Prod QTY 2K | 2/1/98 ≤ $7.95/EA |

SUBJECT: DATE:

Freeman | | 6/23/98

R-Chip Forecast

| where used | QTY | | FY99 (1) | | FY00 (2) |
|---|---|---|---|---|---|
| MMM | 1 | | 324 | | 1089 |
| Analog I/O | 1 | (294+195) | ~~294~~ 489 | (980+653) | ~~980~~ 1633 |
| ADAT I/O | 1 | | 162 | | 545 |
| TDIF I/O | 1 | (129+324) | 453 | (436+1089) | 1525 |
| Analog Line In | 1 | | ø | | 653 |
| AES/EBU I/O | 1 | | ø | | 218 |
| Firewire Card | 1 | | ø | | 272 |
| FX/DSP | 1 | | ø | | 590 |
| MDM 8 ch Comp/EQ | 1 | | ø | | 590 |
| | | | 1428 | | 7115 |

IVY

Alpha / Beta / Preproduction Forecast

| where used | Alpha QTY | BETA QTY | | 1ST ~~QTR~~ PROD QTY |
|---|---|---|---|---|
| MMM | 15 | 25 | | ~~324~~ |
| Analog I/O (EDI card) | 15 | 25 | (294+195) | 489 |
| ADAT I/O | 15 | 25 | ~~(129+324)~~ | 162 |
| TDIF I/O | 15 | 25 | (129+324) | 453 |
| Subtotals | 60 | 100 | | 1428 |
| Extras (50%) | 30 | 50 | (5%) | |
| Total | 90 | 150 | | 2K |

SUBJECT: DATE:

Ivy Core team Mtg 6/24/98

- PDP process being applied to Ivy
- FCS = April 1999
- Staffing changes: Rob → Dir of Eng
  Dan → Ivy Proj Mgr
  Dan O → E4.S, M.I.T.

---

Tom S. MMM Design Review 6/29/98

Present: Tom, Rob B, Grant, Dan F.

Action Items / Issues:

(1) Add MUX in Inchip for AES/EBU main output so can choose any of the 3 S/PDIF's from the R-Chip.

(2) Discussion about HP/MON3 implementation. Ideally, MON & HP have independent volume controls. In Ivy, HP analog volume not ind. of HP/mon 3 volume control in H1.6/R-Chip.

(3) On block diagram, don't show 2 Jacks for Headphones.

(4) Block Diag - Option Cards have H1.6 & R-Chip reversed

Schem Review

(5) Use E-Cap 470uF cap SMD pkg (CA44X)

(6) Use SMD 7805T from EC2A?

(7) Use XREF attribute on nets that cross pages so get automatic sheet-to-sheet XREFs.

SUBJECT: DATE:

NEW PARTS

① DS 1834 Pwr-up reset circuit
② Altera 6024ATQ144-2
③ 100K-Ω R-pack
④ SDRAM
⑤ VHCT245 TSSOP
⑥ ACT244 TSSOP
⑦ Altera 6016
⑧ CY22902
⑨ VHC125
⑩ Altera 10K10 ?
⑪ MAX216
⑫ CS8420
⑬ IM458 64K×16 DRAM
⑭ IC437 R-Chip
⑮ II 416
⑯ New ADC
⑰ New DAC

⑧ 6 layer PCB vs. 4 layer PCB. 6-layer heavily favored by Brent & Dan, supported by Lawrit, Tom gets to make final call.

⑨ Tom Anticipates being in layout on MMM next week. So does Brent or Composer. Brent has priority

PC632
SK632
etc.

⑩ Need for Tom to finish Block diagrams of processor system.

SUBJECT: DATE:

# IW Eng Mtg. 6/29/98

(Circuit)
- Control Surf. pkg rdy for Ken
- P2AS cost is issue on several components.
  - lit buttons
  - faders, cost up because no volume w/Ensoniq
  - scribble strips $11/ea!
  - encoders
- Latest cost rollup is right under $3K. Assumes optomistic costs.
- Steves working on finding cheaper encoders.
- Negotiations have stalled on control surf - rollup is for $3K. Mc Mike hasn't given thumbs-up to go. Ensoniq not providing funding. What's next?

Steve Thompson
- doesn't have time to design control surf.
- Therefore can't get Ken Y. started.
- Mike P. figures it out.
- Awesome

Ron G →
- MWF meetings to discuss SW status. Many issues uncovered.
- Rack + HEA communication issues being worked on.
- Learned to use pre-compiled headers.

Jim W →
- Library / scene management stuff in MAL.

SUBJECT: DATE:

Orzis A

- Found Emulator clock sync bug, fixed R-chip clock circuit. PLL bypass mode would be broken otherwise.
- Emulation S/PDIF & I2S data coming out. EMU32 data coming out but off by 1 clk.
- TOM Sardoing testing on TRAM engine.
- Carl testing FX engine.
- Found bugs in UP I/F.
- Emulation going very good.
- Fujitsu conference call tom about floorplanning

---

7/1/98 Mike Price's Staff Meeting

SXS Report

Accomplished:

- Control Surface architecture & cost rollup
- Pkg for quote to Ken Y for quote
- Identified consultant to assist with Analog I/O
- EDI interface virtually complete, FCC test underway
- MMM Design Review held, PCB layout can begin early July
- Ops Review Plan completed

Plans

- Sign contracts with Ken Y
- Get things rolling on Analog I/O design
- Layout & fab MMM Main PCB
- Finish EDI electrical design
- Vacations
- Start MMM analog board design

SUBJECT: DATE:

Problems / concerns

- Staffing - Control Surface & Analog I/O module
- Control Surface cost too high.

Slides - Dept

① Org Chart, Roles
② ~~Interdepartmental~~
③ Capabilities, Analysis & Dev plans (checklists)
④ Metrics: Milestones
⑤ Interdepartmental Initiatives: R-Chip Prgm Mgmt.

Prod line

① Deliverables, Tasks.
② Budget
③ Risks / mitigation

8/17/99

DM - Fixed.

Style - Fixed, Sarcastic

Dan - Fixed

http:

SUBJECT: DATE:

8/18/98 Mikes Staff Meeting

- Layoff → People.

---

8/21/98 Staff Meeting, Mike P. on Vacation

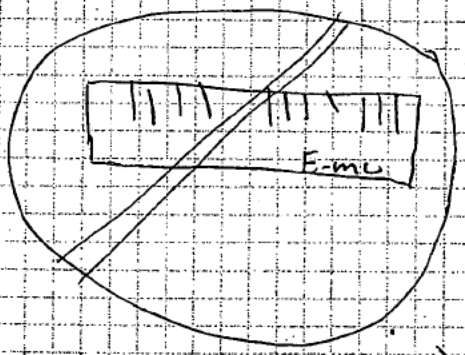
White board:

WE'RE MAD AS HELL, AND WE'RE NOT GOING TO TAKE IT ANYMORE

1. Don't Get out of your office & walk around! MBWA
2. What does Mike Price (not) Do? MBWA
3. We gotta ask the tough questions
4. Marketing provides no direction
5. Product priorities seem exactly backwards
   DTM/Ivy vs. others
6. Why aren't we benchmarking/reverse engineering successful competitors' products?
7. Middle management (Eng/MFG at least) not consulted on key issues & decisions

WE SHOULD OWN HIGH END DTM

Talk to Matt W about DTM mkg.




(No more E-mu keyboards)

SUBJECT: DATE:

8/27/98

EOI Board Monday 8/31/98

Analog Board - Mid (Late) September

October (end) ADAT Board

PCBA Schedule Estimates Ground/Dan

---

8/31/98 Ivy Eng Mtg.

Steve H: Mc Mike is the smartest rat on the ship

What happens if ship control soft late?

- (Mr Mike) Forecast fall off by 80%. Only the ECE customers will buy.
- crippled product.

What if ship with HUI w/ new silkscreen?

Mc Mike: Don't know.

- Additional effort req'd to integrate HUI: 1 month (Rob B)
- HUI should be contingency. Can we mock up new silkscreen for HUI for AES show?

SUBJECT: DATE:

Mikes Staff Meeting 9/2/98

SXS report - Ivy

Accomplishments

- MMM Dig Board designed, fabbed, ready for assembly.
- EDI option card, design review - ready for PCB layout this week
- Contract for Control Surf consultant completed, sent to Ken Y
- Ivy team tried to agree on deliverables possible for Q4 1998 - Assumption is that Control Surf development handled by people outside the Ivy team
- Dan F assigned to Analog I/O module

Outlook

- EDI card layout, fab, Assy - mid September
- MMM Analog Board design, layout, fab - late September
- ADAT option card - late october

Challenges

- Control Surf. Very distracting to Ivy team
- Focus. Need to have each Ivy team member focus on specific deliverables.
- Schedule. April 1999 seems unlikely

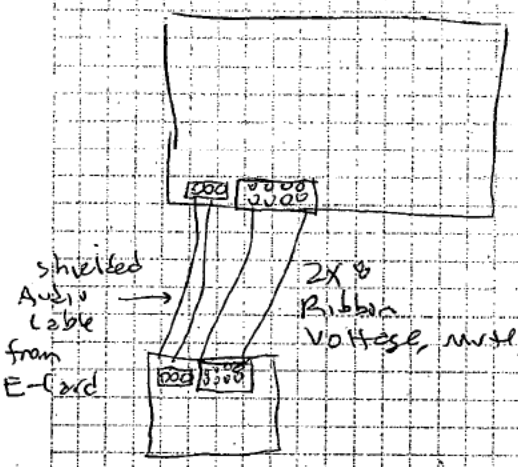
SUBJECT: DATE:

IVY MMM Analog Board Design Review. 9/4/98

Tom, Mil, Neil, Dan F, Steve S.

1. MMM System Block Diagram

→ Headphones distribution - Need to prototype. }




2. Analog Board Block Digram

- Pads for caps on headphone cable & output jack for FCC compliance.
- How to accomodate different impedance headphones? don't try.
- Use E-Card / E-Drive circuit.
- Fix MMM block digram to show correct Headphone topology, 24 vs 20-bit DACs & ADC.
- Use buffers to drive ADC signals to MMM Digital Board (VHC)25

3. List of new components

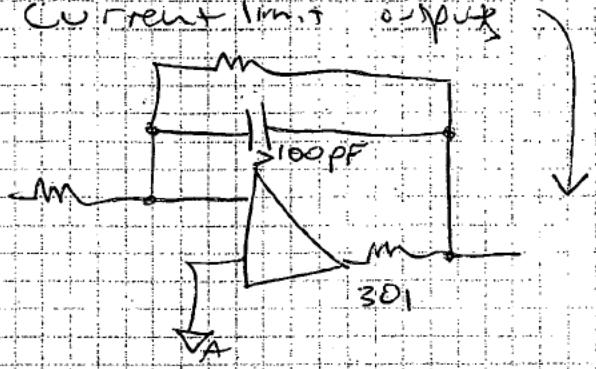
→ Use PCM1716 DAC. → Need to measure true Dynamic range, A-weighted, need 96 dB

NO
Want to use CS4390 DAC $5.50? $13.50 to UMC

- Get Quote on CS5360, CS4390.

SUBJECT: DATE:

- use homemade differential output Drivers.
  Current limit outputs




- use 5532 S08 op Amp
  Philips
  Rohm
  NJR (composer)
- APC 5360
- Want Blue LED for Power.
- VHC125

A. Schedule Activities by other Depts

| | DUE DATE |
|---|---|
| ~~DASK~~ Component Sourcing / Quotes / release to Rev A: (Steve Spina) | 9/10/98 |
| Schem Design Review ~~Feb~~ THM | 9/18/98 |
| Placement: 9/14 (Tom) | |
| Layout: 9/16 (Danny) | 9/21/98 realistic |
| Fab: 9/29 (Whitney) | |
| Assy: 10/5/98 (Jeannie) | |

SUBJECT: DATE:

9/10/98

Ivy MMM Analog Brd Design Review

Nick, Brent, Tom S, Dan F, Nil, Tony D.

1. Prev Action Items

- still need to fix BIK Dizy to show Coax Cable to HP PCB not Ribbon
- → QUOTE CS4390, CS5360 (white may) steve s.
- Use unused vhc125 to buffer RSTN to DACs.
- Add -5V regulator

Schem Review

See SH1

- Use SMD 470uF on +8VA, -8VA C2, C3, C4
- Use SMD Test Points.

SH2

- Add test points
- Stuff=N on BAV99's
- All passives use 0.1uF bypass caps not 1uF.
- Put symbol name or P/N on Dual Stacking Jack
- Revisit gain on inputs. Draw Gain Chart Put on WEB

SUBJECT: DATE:

SH 3:

- use 0.1uF bypass
- Visit calibration circuit for CS5360
- R163 should be 1K
- use R-pak for R156, R161, R162

SH 4:

- 0.1uF bypass caps
- swap Tip & Ring on J2
- Fix Gap connections for C26, C27, C31, C23.
- Nuke R164
- Need GND on D17, D18

SH 5, 6, 7:

- fix as per SH 4

SH 8

- Mod HP topology - 301 Ω, 100P as per other outputs
- Use Shrouded header as per E-DAC for CN4

SUBJECT: DATE:

SH2

- use shrouded header for CNS as per E-mail
- check for L/R swap on HP jack
- Fix/check Depop - See Brian H.

Schedule

- Same sched as prev.

---

9/1/98

EDI Board Fab

| | | | |
|---|---|---|---|
| 10 boards | 3 day | $2180 | |
| | 5 day | $1980 | |
| | 7 day | $1730 | |
| 12 boards | 3 day | $2217 | |
| | 5 day | $1912.68 | $159 |

Order 15 (Whitney)

---

SUBJECT: DATE:

9/14/98 Ivy Eng Mtg

- Add Tom Criswell (Pgm Mgr) to email list
- Add Steve Spira
- Scott Ruda
- Thursday 4PM ? @ Ivy Presentation to Mike, Don, & Eng.

AES: (Mike Barnes)

- Ivy shown twice on Sun & Mon.
- Mantis (my asshole never tells its {source} {smell})
- Press release Friday

Objectives:

- establish real product identity
- Name systems & components, prices
-
- Hype key technologies
- Highlight Surround Sound w/ Tim Burton animation
- Press release.
- 15-minute Multimedia Demo

---

9/14/98

Steve T, Rob B, Dan F.

Steve: Do we really need HUI if we're committed to delivering the control surface?

SUBJECT: DATE:

9/16/98

# Mike's Staff Meeting

- Salary freeze for Mgrs ends Jan 1, no other conditions.
- Stack ranking of engineering staff - next week.
- APS: Ramping up. 250, 350, 500 last month.
- New version for Creative for Q3. Soundblaster studio.
- This month - 2M - 2.5M revenue combined E-mu/Ensoniq Qtr Revenue - 6.9, Combined. Break even is about 2.5M per month.
- APS Euro - need to ship! Might double sales of APS. Need new box, Dennis is Mgr for Box.
- APS Lite might be cancelled. Might do APS for Creative instead. Direct-X, WIN-NT, announcement at Comdex.
- Composer - Proteus 2000, 120 Voice
- Rocky Ultra - Nov? (ETS)
- Audity update: Mktg has proposals to boost sales. only if sales pick up to at least 160 this month.
- Must move Ensoniq Mfg here - adds $1M to bottom line. Our parts cost, overhead lower.

SUBJECT: DATE:

## IVY Deliverables & Personnel Assignments (from 8/11/98)

| | |
|---|---|
| Tom<br>Rob<br>Steve H. | IVY Main Rack |
| Grault | ADAT I/O Card<br>TDIF I/O Card<br>EOI I/O Card |
| Dan | Analog I/O Module |
| Jim | Host Program MAC |
| Ron | Host Program PC |
| Steve H | FX Plug-Ins |
| Scott | HUI Support / Integration of C.S. with MMM |
| Ron | Dynamic Automation |

9/16/98

Chris Neal - Altera.

Floating Nodes: No sweep of current guard.

We have 6 Guards, we'll get to keep them.

Corporate Site license -

$10K/year, irregardless of how many keys/nodes we have.

if sign up for 10K/yr, we'll get 50% more keys for free. (currently keys: Chris will try to sub floating nodes)

SUBJECT: DATE:

last key exires 10/31/99. → we would owe $1K then. write check for $10K instead,
on 9/30/99, get 3 floating licenses for free.

5 of 6 keys are out of date, and can't target new devices

P.O. to wyle. by 9/30?

10KE (9.0 sw) 10K50E comes in QFP

2K x 1 ram blocks are now 2K x 2, dual-ports.

2.5V core, I/O are 5V tolerant on outs
3.3V tolerant on inputs.

Cheaper than 10K50V.

Buy 10K50Vs for 1yy

| | | | |
|---|---|---|---|
| 10K50V RP240-4 | $60 | 100/mo | Budgetary |
| 10K50E RP240-4 | $50 | 100/mo | |
| 10K30A RP240-3 | $25 | 100/mo | |
| 10K30E RP240-3 | $25 | 100/mo | |

Zephal: 400K gate part, 10K LE's,
108 2K x 1 D.P. RAM blocks.
Or each RAM block can be
16 macrocells.
$1200 QTY 1

Up to 2M gates. New tool suite - Incremental compiling, Hooks to Model tech.

CREATIVE
CREATIVE TECHNOLOGY LTD

CONFIDENTIAL

SUBJECT: DATE:

9/21/98

Ivy Eng Meeting

Present: Penny, Tom C, Jim W, Mike B

Late: Tom S (Unprepared) Steve H Scott R Rob B, Steve T. Girault. Ron G.

Rob: AES show Preparation. Ron: Automation. Rack FW. Hot plug-in recovery, Serial cable com. Steve H: Layer above K-Chip Driver. Cont-count on MTC ready for AES. Mtg w/ Tom Marc Jot about Surround Sound, (Environ Audio) When do we need MAC version? Week AMs

Mike B: After mtg, can see Demo by Kevin M. 1 min automation Demo shown at AES. Another station at booth w/HUI. 3rd station has EOS setup, will also have MAC HEA.

Steve T: M² racks Friday. J-F. Stamps due Analog I/O module. Option cards Mech design Done. Front Panel Designs Pending

Tom: EQ1 Fab comein, Kits Finished. PCB35 Analog Brd out to FAB, back monday. Kitting in Process. PCB40 front panel board in layout. Main Digital board. Grinding Halt due to [redacted] equipment completion. Problem with SysChip?

Dan F: Task lists. Yes: Girault, Rob, Dan F, Control Surf. Needs task list, Tom S not done.

CREATIVE
CREATIVE TECHNOLOGY LTD

CONFIDENTIAL

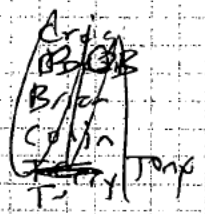
SUBJECT: DATE:

Steve H: Anyone going to Control Surf Design workshop at AES Tuesday 2pm?

Open discussion: Schedule. Equipment comparison.

Mikes staff mtg 4/23/98

| | | |
|---|---|---|
| Lisa B | 2 | 3 |
| Mark M | 3- | 1.5 |
| Danny G | 5 | 6 |
| Jeannie M | 1 | 4 |
| Brian H | 2+ | 3.5 |
| Mark C | 1 | 4 |
| Susan F | 1 | 4 |




| | |
|---|---|
| Craig | 1- |
| Bob | 2 |
| Brian | 2 |
| Colin | 5 |
| Tony | 1 |
| John F | 1 |
| Terri | 2 |
| Ron | 1 |
| Gireult | 5- |
| John M | 2- |
| Scott | 2 |
| Tom | 5- |
| Duane | 5 |
| Mel | 2+ |
| Jim | 1- |

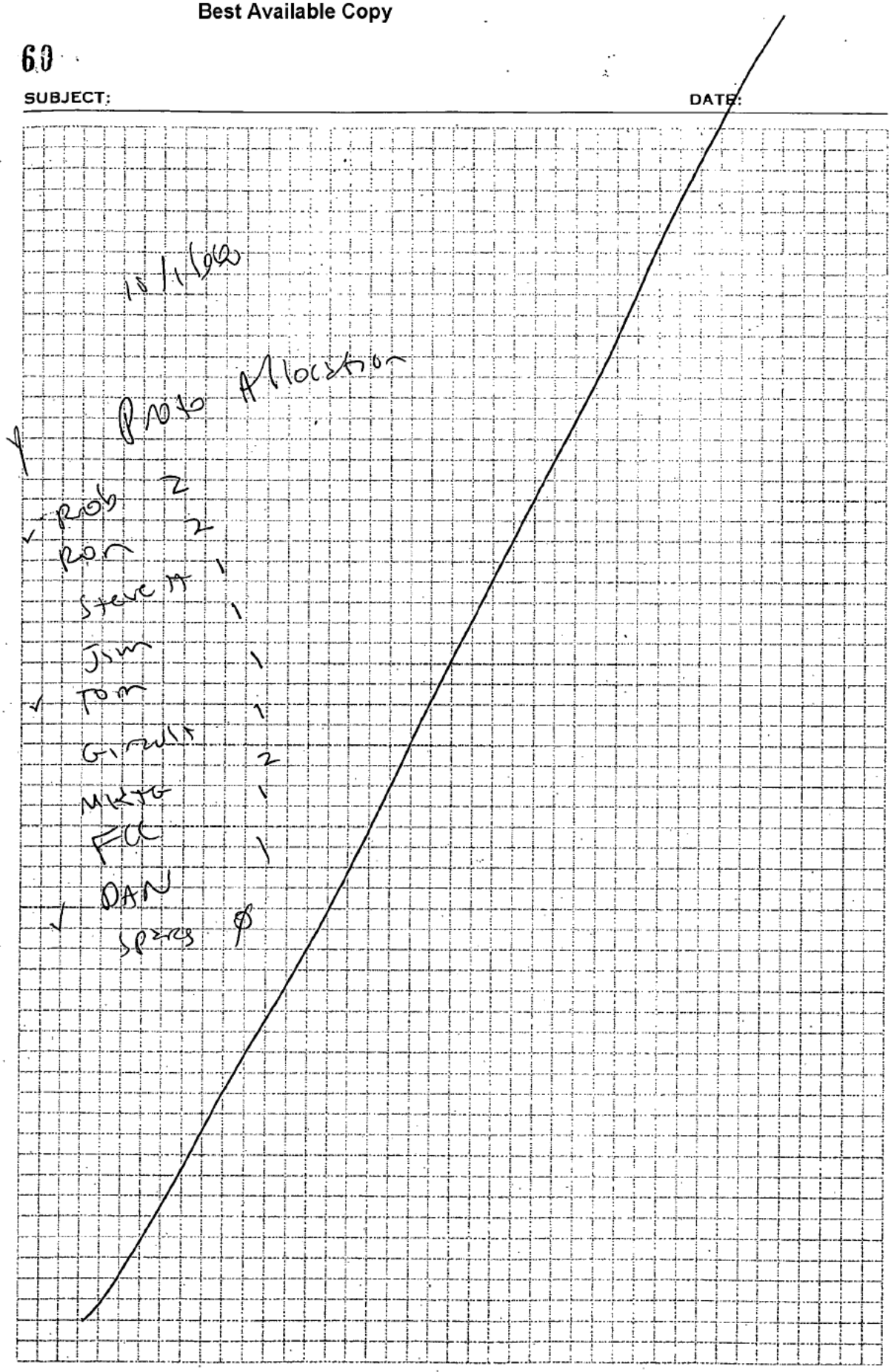
SUBJECT: DATE:

10/1/99

Photo Allocation

Ivy

- Rob 2
- Ron 2
- Steve M 1
- Jim 1
- Tom 1
- Gizmo 1
- Mktg 2
- FCC 1
- DAN 1
- Specs ?



SUBJECT: DATE:

10/1/98

Ivy Schedule Preparation - Notes

1ST Pass schedules:

→ Rob B "May", includes some beta testing
→ Steve T: May, unscrubbed, 40% confidence, includes beta
→ Girault: 8½ months, not including beta testing
→ Tom: 7.3 months, not including beta
→ Dan: 7 months assuming F/T, includes beta
→ Brians: Tape-out Tuesday NOV 1 samples. Fujitsu manpower problems, Layout problems. Risk factor: Fujitsu might take on 2-week turn.

10/5/98

Ivy Eng Mtg. 9:

SW Status: Rob · AEI preparation

- SSOT port going well. Cache, SRAM, LEDs work.
- Steve: Rchip layer for FX System.
- Scheduling
  next week: Mtgs on Environmental Audio. Automation engine Design Review MAX FIL I/O (works!)

Mech: Steve T

- chassis to: everything fits (MMM)
- Analog I/O module work

SUBJECT: DATE:

Tom S HW

- Computer/clk sys works
- Next: Inchip debug, make Audio path work
- Circuit: Inchip
- EDI CARD ready for Integration
- Analog boards in ASSY done today
- Circuit starting ADAT board design
- A Clock diagram has been done, ur web soon.
- When will there be a SMPTE System diagram? See Circuit
- Audio path, Clock/Sync, Microprocessor block diagrams in progress by Circuit/Tom

Dan: Schedule

- Tom C is Schedule man
- Andre will do TDIF CARD
- SW in trouble
- Inchip in trouble
- Tom C will keep Schedule current bi-monthly
- 2 weeks from now: Tom C initial Schedule

Mike B: AES

- Nominated for Blue Ribbon award by EQ magazine
- full write-up in November
- SAL and Ray from Ensoniq are newly interested in the control surface
- Got another HUI from Mackie