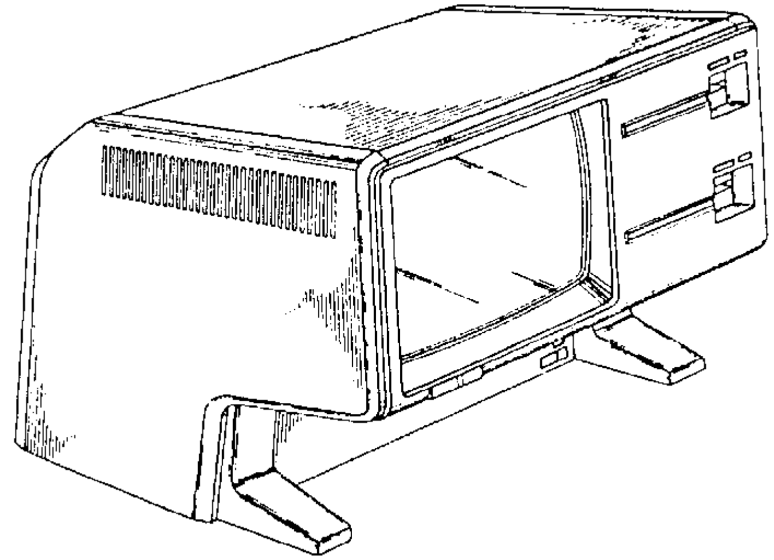
Apple Lisa Computer Historical Information

Apple Lisa Computer Desktop Manager Info

David T. Craig
29 December 2000

Email sent to the LisaList mailing group.



From: David Craig
Sent: Friday, December 29, 2000 11:39 AM
To: 'lisalist@mail.maclaunch.com'
Subject: Apple Lisa Computer Desktop Manager Info

Apple Lisa Computer Desktop Manager Info

Compiled by David T. Craig
29 December 2000

Here's some information about the internal workings of the Lisa Desktop Manager from the early 1980's that may interest readers of the LisaList mailing list.

The Desktop Manager was the program that the Lisa ran at startup and which gave the Lisa its "desktop metaphore" user interface. This program can be considered equivalent to the Macintosh Finder program, but it wasn't until Macintosh OS 7 was released in 1990 that the Macintosh's Finder started to have a feature set that was close to the Lisa's Desktop Manager of 1983.

The Lisa Desktop Manager was implemented by Dan Smith (now called Dan Keller) and Frank Ludolph. Many others had a hand in its design including Bill Atkinson, Larry Tesler, and Bruce Daniels. For a great description of the development of the Desktop Manager see Dan Smith's March 1986 interview in the Lisa magazine Semaphore Signal titled "The Past, Present, and Future of the Macintosh Desktop". Semaphore Signal issues can be found at the following web site:

http://www.semaphorecorp.com/ss/toc.html

This interview can be found in Semaphore Signal issue # 26 dated March 1986 at the following web site:

http://www.semaphorecorp.com/ss/ss26.html

At the end of this message I have included the full interview.

From a technical perspective, the Lisa Desktop Manager was just a Lisa "shell" program. A shell program was basically a regular Lisa program but with a name starting with "SHELL.". At startup the Lisa ran a program called the "Environments Window" which let a user pick a shell to execute. This environments window normally automatically picked a shell if there was only one installed on your Lisa's hard drive, or if the Lisa user had previously specified a default shell. Programming a Lisa shell required a few extra features that normal Lisa programs did not have since shells had to communicate with the

Lisa startup and shutdown processes directly. Apple created a technical note describing these special features.

The Lisa Desktop Manager is best described by Apple in the following documents which I have quoted in this LisaList message:

o The Architecture of the Lisa Personal Computer
Bruce Daniels
Apple Computer, 1984

o Lisa User Interface Guidelines
Chapter 5: Desktop Manager
Apple Computer
November 1983 (alpha draft)

Other sources of Desktop Manager information are the Lisa Owner's Guides. The Lisa 1 and Lisa 2 owner's guides from 1983 have great chapters on the initial versions of the Desktop Manager. The Lisa Office System 3.0 manual from 1984 describes the final Desktop Manager version which supported features such as the Desk menu which was somewhat equivalent to the Macintosh's Apple menu.

Other historical pieces on the Lisa Desktop Manager that are worth reading since these are from Apple Lisa team members are:

o The integrated software and user interface of Apple's Lisa
Edward W. Birss, ca. 1980s

o An Interview with Wayne Rosing, Bruce Daniels, and Larry Tesler
BYTE Magazine, February 1983

o Inventing the Lisa User Interface
Frank Ludolph, Rod Perkins, Dan Smith
ACM Interactions, 1997, vol 4, issue 1, pp 40-53

This Inventing paper is great since it shows samples of screens for the Desktop Manager from 1979 to 1984. This paper can be found on the following web sites:

http://developer.java.sun.com/developer/techDocs/
hi/papers/Lisa/Interactions_96.html

http://home.san.rr.com/deans/lisagui.html

https://secure.acm.org/pubs/citations/journals/
interactions/1997-4-1/p40-perkins/

A short paper on the Lisa user interface can be found in the following article:

o The Lisa User Interface
Frank Ludolph, Rod Perkins
ACM SIG-CHI 1998

This paper can be found on the web at:

http://dev.acm.org/pubs/articles/proceedings/chi/286498/
p18-ludolph/p18-ludolph.pdf

For a great discussion of the philosophy behind the Lisa's user interface see Frank Ludolph's comments that he presented at demonstration of the Xerox Star computer that can be found at the following web site:

http://www.stanford.edu/~hodges/Xerox/ReStarDemoAncmt-5-FrankLudolph.txt

I have included this talk at the end of this message.

The author of this message (David Craig) also wrote a short paper on the Lisa GUI for college:

o Apple Lisa Graphical Object-Oriented User Interface
David Craig, 1987

For a wonderful discussion of the origins and changes in the field of GUIs see the following paper which contains a series of comments about this area during a demonstration of the Xerox Star in 1998 by many former Xerox Star people:

o Xerox's Contribution to the Development of Office Automation
via the Desktop Metaphor and Distributed Computing Systems

This paper can be found at the following web site:

http://www.stanford.edu/~hodges/Xerox/

Here's one other piece of Lisa Desktop Manager trivia that may interest current Macintosh users. If you hold down the Option key while in the Macintosh Finder and select the Apple menu, you will see a menu item labeled "About the Finder". Selecting this menu item displays a dialog with a picture that lists the people behind the Macintosh Finder. You can even do this with Macintosh OS 9 which was created in 1999. Wait a while and you will see the following credit to the Lisa Desktop Manager:

Lisa Desktop Manager
1981-1983

Dan Smith
Frank Ludolph
Bill Atkinson

For a good description of how the Lisa's software in general worked see LisaGuide, Apple's interactive tour ofthe Lisa. I have a set of screen dumps from this program that are useful if you don't have access to LisaGuide.

One other piece of Lisa Desktop Manager history is that the Macintosh Finder influenced the design of the Lisa Desktop Manager. Per Bruce Horn of the Macintosh Finder team, he showed a mock-up of an object-oriented Macintosh Finder from 1980 or 1981 to Bill Atkinson who was working on the Lisa team at that time. Bill used the Finder mock-up to devise an object-oriented GUI for the Lisa that had at that time a less interactive dialog-based interface.

"The Architecture of the Lisa Personal Computer" has the following Desktop Manager description in the section titled "The Lisa Destop Manager and Applications":

"The Lisa Destkop Manager serves the same basic functions as the Shell or command interpreter in conventional systems. It provides a mechanism for the user to create and manage documents or files (copy, move, rename, delete, etc), to run tools or programs, and, in general, to control the system. The desktop image, implemented through the Window Manager, is treated as a special window that is always open to the full width and height of the screen, is always behind any open windows, and has a grey background pattern rather than the usual white. The Desktop Manager displays the icons that are out on top of the desktop and the icons in any open windows assocatied with disk, diskette, or folder icons. The Desktop Manager recognizes the user's manipulations of any of these icons and responds interactively with the appropriate visual feedback. The Desktop Manager also performs any filing operations such as file deletion or copying implied by such manipulations, invoking the necesasry OS file system calls, and then displaying the resultant visual image. When auser "opens" any document icons, the Desktop Manager first determines the exact type of document which the user desires to open. Associated with each document type is a Lisa tool, or application program. The Desktop Manager then creates an OS process running the desired tool. Next the Desktop Manager calls the Window Manager to establish a window with the same size and position on the screen as the document had when it was last opened. The Desktop Manager sends a DocOpen event to the new process passing both the window to be used and the identity of the document to be opened. When the application process receives the event, it opens the document files and displays the document in its window. Finally, the Desktop Manager makes the new window be the active window so that the user can proceed to manipulate and edit its contents."

Note that the above discussion refers to processes and events. The Lisa OS supported multiple processes and inter-process communications called "event channels". The Desktop Manager itself was just another process.

"Lisa User Interface Guidelines" has the following Desktop Manager description (I've only listed parts of this info since it is basically a listing of key aspects of the Desktop Manager that Lisa programmers using the Lisa Toolkit framework need to be cognizant about):

"General scheme:

Some applications limit the size of a document to one whose contents can be kept in memory at the same time (for example, LisaCalc). In these applications, any changes the user makes to the document are kept in memory. When the user saves the document, the changes are made to the document on the disk. If the user chooses Revert to Previous Version, the changes are nullified.

In other applications, the user can create documents that are too large to fit into memory (for example, LisaList). In this case, the application keeps a file containing the changes to the document. When the user saves the document, the application updates it based on the material in the change file.

Deskop Manager operations:

Activate: The active window is the one the user is currently working on. At most one window on the desktop can be active at a time.

Deactivate: The user deactivates a window by clicking outside it.

Set Aside: When the user sets aside an open document, the Desktop Manager deactives it, and its window is shrunk back to the original icon.

Save & Continue: Save & Continue saves all changes to the document without closing its window.

Save & Put Away: Deactivates the document, saves its contents, and returns its icon to the location of its ghost.

Suspend: Suspends a document that is open or set aside in the following cases:
1) User turns the power off on the Lisa.
2) User ejects a diskette that contains the application, part of the document, or both."

"The Past, Present, and Future of the Macintosh Desktop" in Semaphore Signal issue # 26 from March 1986 has the following to say about the Desktop Manager's origins and development:

"The Past, Present, and Future of the Macintosh Desktop

To a first-time user, perhaps the most striking thing about the Macintosh is its use of the desktop metaphor: the folders and other icons intended to help make the Macintosh a user-friendly machine. For a perspective on where those ideas came from, how they were further developed by Apple, and what they might lead to in the future, we interviewed Dan Smith, an Apple Principal Software Engineer.

Signal: Give us a brief history of your career at Apple.

Smith: I've been at Apple for a little over five years now. I initially signed on to the Lisa project to work on what we called the Desktop Manager, essentially the equivalent of the Finder on the Macintosh. I worked on that for about two years, until the whole Lisa project was near completion. Then I became User Interface Coordinator for the Lisa project, then switched to a consulting role for the Macintosh, since Mac picked up about halfway into the Lisa development stage. I took some time off from the Macintosh and Lisa to start working on some future projects, did that for about nine months, then got pressed back into service to do a program development environment for the Macintosh, which is what I'm working on right now.

Signal: Were you the desktop programmer? What was the organization responsible for the desktop and the other Lisa software?

Smith: The effort was split up into a couple of different groups. There was the desktop group. Two of us actually did the implementation. I programmed the user interface portion, and Frank Ludolph did a fair amount of the lower level implementation. Then there was the applications group, and that was split up into essentially the different applications that came out: LisaDraw, LisaWrite, and so on. There was also an operating system group, which did the much lower level software.

Signal: How did the ideas for the desktop originate, and how were they incorporated into your design?

Smith: That's a pretty interesting story. When I started at Apple, the idea of the desktop hadn't really quite been born. In fact, it was thought we'd do something fairly simple, and it would be a one-person job for a couple of months and it would be over. It was a little later in the project that we realized the desktop was going to be a central part of the entire system. The idea of an iconic form didn't come along until quite late into the development of the product. We started off with something that was pretty Smalltalk-like. There was a notion of a thing called a browser, which is essentially a table you could flip through, listing the documents you had in a hierarchical fashion. But the whole initial desktop was essentially technically oriented. We went through iteration after iteration. I remember doing prototype after prototype, and trying them on several groups of people, getting it to be more and more useable. But a number of us were not happy with what we were getting, so fairly far into the project a couple of us took a radical departure. We took a fair amount of our own time developing an iconic model, then sprung it on the whole group. It met with some resistance, but the majority of people really liked it. Then it was a mad rush to incorporate it into the final product.

Signal: Exactly who did what? Was the design fluctuating with the whims of a few individuals?

Smith: The desktop had a pretty fiery history. We did have design reviews of all of the components of the system. We had teams: writers and marketing and engineering people who were all involved actively in the product. When we were actually doing the design and programming the application, the specification circulated amongst the teams. They all had a voice in the initial design of the product. As we circulated some of the actual design, and showed some of the prototypes, there was some frustration by some people that the desktop was not as easily useable as we would like. A couple of people, myself and Bill Atkinson primarily, and later Bruce Daniels, snuck off and worked on a prototype iconic model.

Signal: Where did your ideas for the iconic desktop come from?

Smith: It stemmed from a number of different places. From some of the work done at Xerox on the Alto computer. There were some IBM research papers, believe it or not, that discussed some office models that were iconic based that we looked at. And somewhat after we had worked on this, the Xerox Star computer had come out, so we compared it to our model. There was also some work done at MIT that had some influence on us. We got together and kicked around a lot of ideas, and tried to keep it consistent with our user interface philosophy. After a short period of time, we put together a prototype that stayed pretty close to the final product.

Signal: Was the big breakthrough essentially just realizing how to use icons inside of windows to represent files? Were you already using windows and pull down menus?

Smith: The initial design already had windows and pull down menus, although even those had some history. But the actual desktop's primary function was a filing function, an organizational function. The iconic model was really the late development.

Signal: So various ideas would come, they'd be implemented and tried, and then approved by peer groups, as opposed to steadily working toward some predefined specification?

Smith: That's not quite true, although the desktop was the exception to the rule. All of the major software products were designed by a group responsible for that component, and were pretty much spec'd out completely before any implementation was done, although there would be little mock-ups to check out some of the ideas. We tried to do the same thing with the desktop, but it's one of those products everybody is forced to use, and everybody has an idea on how it should be, so there was a lot of feedback about that particular component. It took a fair amount of iterating to come up with something that appeared to be not only satisfying to the groups working there, but also something that went over fairly well in our user testing, which we did quite a bit of. We tested mainly on people who had very little computer experience. We tried initially to make it mainly those who were in Apple, who had very limited experience, maybe new hires or people whose job didn't involve working directly with a computer. Further on into the project, we actually hired some people for testing purposes and we had a set of people in-house whose primary job was to gather testing feedback. We even set up a couple of testing rooms, and carefully recorded everything that happened, and tried to draw whatever conclusions we could. It took a fair amount of that to get us headed in the right direction.

Signal: What are some examples of things considered for the desktop, but that ended up being changed or left out?

Smith: The initial version had icons which were three dimensional and much more cartoon-like and somewhat entertaining. In fact, the initial trashcan was this beat up old trashcan you'd expect to see in an alley, with the lid half open and flies buzzing around it. We had actually talked about putting in some sound effects for when somebody put something into the trashcan. That met with some resistance from some of the stodgier people on the team, so that was dropped. One thing in particular we had a heated debate about was the notion of whether a document had to be saved or not. A number of people on the project wanted the system to be as far removed from typical computer interactions, and be as concrete as possible. The argument was that when you write something down on paper, it's always there. You don't have to say "save" to prevent it from mysteriously disappearing, as opposed to a typical computer model where people think "I know I have something in the computer's memory, and it's temporary, and I have to make sure I get that information from memory onto the disk otherwise I'm going to lose it". So there was quite a heated debate on that one, and unfortunately we tended to battle more toward the computer model, for a couple of different reasons. One reason was familiarity, the idea that users who had computer experience were more used to saves. There were some technical considerations too, having to do with how much information we had to keep around all the time when somebody stopped working on a particular document. Another desktop feature was the problem you always have anytime you go to a filing system with folders buried in folders. This was something Frank Ludolph was very concerned about. How do I find my documents, how do I find my folder? With folders nested in folders, and having to double click and open them and search around, it became quite tedious to find something if you had misplaced it. Rightly so, he thought we were duplicating some of the frustrations a person would find in a normal office, having to manually dig through their filing cabinet to find something. A computer should be able to take care of that job for them, essentially doing an electronic search of the filing cabinet. That was a feature we wanted to implement on the Lisa, but didn't quite have time to do.

Signal: How would automatic searching have been accomplished?

Smith: That feature wasn't fully specified, but it would be primarily by specifying certain attributes about the item you were looking for, whether it was simply the title of the document you were looking for, or some key words in the document, almost like a database query. Something to make a search a lot less painful.

Signal: What other desktop capabilities never made it into your final model?

Smith: One was the ability to allow new applications to be added, which could operate on documents not of their type. Suppose somebody wanted to add a new compare program that compared two text files for example, and gave you the differences. With the user interface model we had, simply clicking on a document opened that document, or clicking on an application opened that tool. There was no real clear way of specifying documents of different types as essentially parameters to another program. Probably the biggest single feature missing in both the Lisa and Macintosh is the ability to automatically remember a series of actions and play those over again. One of the things the desktop did was to make most operations very simple to do, which is a great benefit to first-time computer users. But computer users expect the ability to give the computer a series of commands and have the computer record those, and save them from the drudgery of having to repeat those over and over again. Primarily because of not having quite the right user interface for it, and the technical problems involved, that particular feature was never fully implemented in the Lisa or the Macintosh.

Signal: Were there any schemes that would allow repeatable commands?

Smith: There were a couple. One that comes to mind that was actually done with limited success was to simply record all the actions of the user on a real low level: the user moved the mouse from here to here, clicked the button, typed keys at this point. That allowed us to do little demos where the machine appeared to be running under automatic control. The problem with that approach is it requires you be in the exact same state every time you run this little script, even down to the exact position that icons are on the screen or within their folders. That situation turns out to be very rare, so that approach is flawed. It's also not something you can easily parameterize. You pretty much specify an absolute action.

Signal: Programmers were disappointed to find that this beautiful, iconic, straightforward, user-friendly, state-of-the-art desktop was built with a fairly old programming language and tools. The Macintosh didn't come with any new, powerful systems to make programming a Mac as easy as using one. Is there a lack of tools for developers to easily implement new Macintosh programs?

Smith: There's no question that implementing a Macintosh type of user interface is difficult, and the initial implementation on the Lisa and the Macintosh was done in pretty traditional form. What we've tried to move towards is more of an object-oriented implementation that involves extensions to existing languages. In the long run, that's really going to be the way to develop software for this type of a system. The problem you typically have with object-oriented software is performance and size. Since it's a general system, you tend to get a lot more software than you actually want because everything is built in, and you may be only using a subset. The size of any application tends to be quite a bit bigger, and the performance often can be quite a bit sluggish.

Signal: Has object-oriented programming really been proven? Both the Lisa and Macintosh have demonstrated excellent user software, but only using traditional, non-object-oriented programming. Apple's object-oriented systems always seem to arrive a day late and a dollar short and have never really been used to implement anything of commercial significance.

Smith: That's certainly been true up to this point. Whether you can use that approach or not depends on how much horsepower you have behind you. Smalltalk is heavily object-oriented. Depending on what kind of machine you're running on, Smalltalk is a wonderful system to use, or it's just horribly slow and painful. So it's a combination of the hardware you have, and the software implementation. Programmers who use an object-oriented system are really pleased with it, because of the amount of work that's already done for them. Developers will gravitate towards that type of a programming solution once it's implemented in the most efficient possible way, and there's sufficient hardware behind it to make the performance acceptable.

Signal: Why were the tools that Apple used to implement its first desktop applications never released to developers?

Smith: Great question. The reason is we didn't have the object-oriented extensions made to any of our languages at the time we were developing the product.

Signal: If the Apple programmers didn't need finished object-oriented extensions to complete their applications, why did you think third party developers would have to wait for the extensions to be completed before they could get tools for their products?

Smith: The primary reason was because of the difficulty we all faced while developing the applications. We were all working at Apple on a daily basis, and had access to everyone on the project on a face to face basis, and we were still having difficulties in the amount of time it took to implement just a standard user interface. Each of the applications had to implement scrolling, and the menus, and all of the appropriate things, according to the standard user interface and specification. It turned out to be just incredibly difficult to do, and incredibly time consuming.

Signal: Couldn't Apple programmers borrow each others code and algorithms?

Smith: Well, we tried to some degree, but there would be these slight little differences. It was very difficult at times to trade some of that stuff, because of the way an application was structured. We were essentially pioneering all of those techniques, even the basic design of an application being event driven, as opposed to reading in a little command line and a carriage return and spitting out the answer. It was all quite a bit different from the programming experience of most people there. We just realized way too much effort went into developing desktop applications, and something better had to be done for outside developers.

Signal: How did you get involved with the Macintosh group?

Smith: Towards the end of the Lisa project, Macintosh development was getting much more established. I spent a little bit of time with Bruce Horn and Steve Capps, the people who worked on the Finder, because they were doing the same thing I'd done on the Lisa. We were able to share some ideas. That was useful for all of us. I had already conquered a number of the problems they were facing and, in their development, they had found a couple of interesting solutions to problems I had not found a solution for.

Signal: What's an example of something they found a solution for?

Smith: They had a user interface solution for the problem of how to get a tool to operate on documents not of its kind. The solution was to simply select all the documents and the application involved, and click on one of those as a set.

Signal: Wasn't the Macintosh group reinventing the wheel by not letting you write the Finder?

Smith: It was to some degree, but the Macintosh group was faced with constraints which were pretty extreme compared to those on the Lisa. The amount of memory available and not having any real hardware memory management made some things quite a bit more difficult on the Macintosh.

Signal: Was the Macintosh a step backwards from the Lisa? Was the machine sorely underpowered for what it was trying to do?

Smith: Not with respect to the original design goal. The original goal had envisioned the Macintosh as being a personal computer that would run a single application at a time. It would have superior graphics and the Lisa-style user interface. But it was going to be restricted in terms of exactly how much it could do. It turns out that in terms of computational power, in terms of raw speed, the Macintosh actually exceeded the Lisa to some degree.

Signal: Do you feel the Macintosh group did a good job of capitalizing on what the Lisa group had pioneered? Or were not all the lessons learned?

Smith: Yes and yes. They did a tremendous job of leveraging off what Lisa did. All the graphics routines carried over, almost to the byte, thanks to Bill Atkinson's foresight. The window and menu management routines were essentially brought over as is, at least as far as the interface. The major changes that had to be done for the Macintosh were that things had to be recoded in asssembly language in order to get the program sizes down. But most of the interfaces and stuff like that were pretty much the same, but with improvements along the way. In that sense, the Macintosh was really a second generation Lisa, so they had an opportunity to improve on a lot of things. In some cases, a step was made backwards.

Signal: What's an example of Macintosh taking a step backward?

Smith: A step backward was the "run one program at a time" model. For example, you clicked on MacPaint, the entire screen would be cleared, and up would come something that was totally different than what you had been looking at before. The desktop model disappeared on you, and now you were running this entirely new program. If you wanted to do any type of filing operation, you either had to quit that program, go back to the Finder and manipulate your files, or there was this little dialogue box that allowed you to search through a list of names as opposed to seeing them in their iconic format. On Lisa, you stayed in the desktop the entire time and would just click on another icon to open it. That would open up a window, but you would stay in the desktop. You wouldn't get this dramatic state change, and you also wouldn't have to go from at one point dealing with icons to, at another point, dealing with a list of names.

Signal: How much of that step backwards was forced by the hardware limitations of the Macintosh? Could the programmers have pulled off a multi-application environment like the Lisa?

Smith: That's a tough one. With the 128K of memory available on the original Macintosh, they really couldn't have pulled it off. But with the 512K Macintosh, it was definitely more possible. Now with the Mac Plus, it could be done quite easily.

Signal: There are a lot of little differences between the Macintosh Finder and its predecessor, the Lisa desktop. The Lisa desktop underwent cycles of consumer testing and design reviews. Is it true the Finder programmers were influenced by a lot of Lisa features, but ended up doing many things their own way just because they wanted to?

Smith: That's fairly accurate. The Macintosh group had much more liberty as far as the user interface design. They had the opportunity to make little tweaks to try to improve on the overall design. It wasn't that the Finder was totally unreviewed by anyone other than the programmers, but there was definitely more liberty. That's not necessarily a bad thing. The programming team on the Macintosh was quite a bit smaller. With a smaller group, they had an opportunity to have the product be a little bit more consistent, because you get more of a single-mindedness to the product. It didn't look like it was hatched up by twenty-five different people.

Signal: It's interesting how the Macintosh was able to successfully provide a standard set of programming routines for developers, even though they aren't object-oriented, compared to the Lisa group's inability to do that. Why?

Smith: It has to do with the initial orientation of the project. When I first started on the Lisa project, the goal was to produce a machine essentially not programmable by outside developers. When we later decided it would be appropriate to have outside software developed, we were in a little bit of trouble, because we had created quite a complex system. It was very difficult to program. Whereas the Macintosh software architecture tended to be open at the very outset, and all the routines were carefully documented and carefully designed with respect to outside people using them.

Signal: What was your influence on MacPaint?

Smith: While I was working for awhile on future products, I was working with Bill Atkinson and others out at Bill's home. Bill was still working on MacPaint at the time. We would get together in his workroom and kick around various ideas about the MacPaint user interface. He would cook up a new version and hand it to us, and we would play with it and give him feedback. We got to be firsthand users with that product. That was a real treat.

Signal: MacPaint, for all its greatness, violates a lot of Apple's user interface guidelines. Was there ever any concern about that?

Smith: Yes, there was, as a matter of fact. There were a number of people who were opposed to some of the things Bill did. Bill's feeling was that we needed to pioneer some new ideas, and there were just certain things in some of the user interface guidelines which weren't appropriate for his application. He decided to push for some of those things, and hope for the best. It turns out he was able to move more towards the user interface guidelines down the line.

Signal: Why did it take so long for LisaDraw and LisaProject to migrate to the Macintosh and become MacDraw and MacProject?

Smith: The primary reason was the memory constraints on the Macintosh. Most of the programs on the Lisa were huge by comparison to other software products. Significant recoding had to be done in order to get the products moved over to the Macintosh. It takes a long time to program in assembly language, or to reorganize your entire program with space a primary consideration.

Signal: Didn't anyone realize Apple was trying to get the Macintosh to do Lisa-like things? Couldn't you have just put in more memory and saved many programmer-years of effort?

Smith: A lot of people said, myself included, before the Macintosh even came out, that the thing was designed with memory that was way too low. The primary motivation there, of course, was to keep the price down. They wanted the Macintosh to be affordable to the common person. This was going to be a computer most people could go out and buy. As time went on in the project, more and more features were added. They decided to go with a different disk drive, with different memory, and slightly larger display, so the price ended up creeping up out of the reach of the common person, to the $2500 figure. At that point the memory limitation perhaps wasn't as good of an argument as it was earlier, because at that point the memory wasn't the dominating cost of the product. So I'd have to agree with you on that one. If the Macintosh had initially come out with a larger amount of memory, it would have saved significant amounts of time in development.

Signal: Is there any hope for a true, Lisa-like, multi-application desktop environment on the Macintosh?

Smith: There certainly is.

Signal: You say that like you're hinting.

Smith: It's sort of like a hint. I know of some outside developments in that area, outside of Apple. There are a couple of efforts going on there. I wouldn't be surprised to see something like that fairly shortly.

Signal: Switcher and the desk accessories have always seemed to be somewhat desperate attempts to give the Macintosh a Lisa-like environment. Couldn't Andy Hertzfeld have made Switcher juggle multiple applications on the screen at once? It's already executing any application on demand, but idle applications are kept off-screen. What if the Switcher's multiple screens were shrunk down to icons and saved on a visible background, or occupied windows that could be scaled and overlap? The Macintosh could still jump between applications, and yet the user could see all of them on the screen simultaneously.

Smith: I'm not sure if that particular idea was around at the time, but Andy's initial goal was to just do something simple, and that had the highest chance of success. Switcher seemed a fairly simple idea, although it turned out, because of the Macintosh architecture and the system design, to be quite difficult to do. It really took someone with Andy's intimate knowledge of the low level aspects of the system to pull that off. However, now that Switcher has been done, and it's been shown it's possible, we will see more Lisa-like models, where the visual continuity is retained.

Signal: Where do you see the user interface going? What will future products look like?

Smith: What we'll see in the long run, in terms of software products anyway, is a much tighter integration. You'll be able to buy small tools that fit into the rest of the system and work in combination with all other tools. Say, for example, you're trying to compose a memo to someone. In front of you is your piece of paper in electronic form, and you're typing or writing on it. You want to do something like check the spelling of a word. Rather than necessarily invoking the spelling checker built into the word processing program you're using, you simply use the spelling checker you purchased the other day. It would be tightly coordinated with the overall system, so you could take that tool or any other tool and use it to operate on the document you're currently working on. What it's going to take to do something like this is a real solid low-level system design that provides a common data structure format everyone can plug into and that provides very easy access to all data on the system, whether it's a picture or a document or a database or whatever. Some sort of uniform style, such that any tool that's around can be used on any document. Some of that has been done to some degree with systems like Smalltalk, but nothing in any commercial sense.

Signal: That's a major hurdle for developers to overcome, but it probably won't seem like much of a breakthrough to users. How will products change in an obvious visual way that would be apparent from, say, looking at an ad for a computer of the future?

Smith: I don't think mice will be around forever. We'll move more towards something that allows you to bring the desktop metaphor closer to home, closer to reality. You could imagine your office desk actually being a display, and you actually manipulate information right on the desk.

Signal: What gets rid of the mice? Voice?

Smith: Not necessarily. You may be writing on the desk as you do now, with a pen of some sort, or using a lot of the tools you would use today. They might be touch sensitive to some degree, such that you could move things around by touching them. We're looking more and more towards making the abstraction more and more concrete. Voice has its place, but not as big a role as some people think it does. People will find they don't really want to talk to their computer. Not only that, but it's not very efficient. Imagine a voice driven car and telling it to take a left turn, but not specifying quite enough detail.

Signal: If you had the super deluxe model, all you'd have to say is "take me to the airport". Can you somehow characterize any of the work you've actually done on future products?

Smith: The only thing I can really say is that it's been looking down the line five years or so, with respect to what hardware will be available. A lot of the ideas people have had, Alan Kay in particular, who is now an Apple Fellow, require a significant amount of hardware. Some of those ideas are probably going to see the light of day in the not too distant future, as the hardware developers catch up with the minds of the software developers."

Frank Ludolph's comments about the Lisa user interface that he gave at a demonstration of the Xerox Star in 1998 are:

"Date: Thu, 4 Jun 1998 12:00:20 -0700 (PDT)
From: Frank Ludolph <frank.ludolph@Eng.Sun.COM>
To: Multiple recipients of list <exxerox@whyanext.com>
Subject: Re: The final live demonstration of the Xerox 'Star' computer

Jeff mentioned the CHI joint Star/Lisa demo. As one of the presenters it was a blast to do. Most of you have experienced 'Oh, I should have said...' Jeff's comments give me the chance to say it, though to a differenct audience. :-)

> From: jeffjohnson@igc.apc.org
> …
> 2. When Frank Ludolph was giving the formal Lisa demo, he edited a graphics
> file. When he tried to close the file, a dialog box appeared, reading "Do
> you want to replace the document that you last edited 7 years ago?" The
> audience got a laugh out of this, but I think it was quite noteworthy for
> two reasons, one of which Frank mentioned and one of which he didn't.
> Frank said that the document had in fact been last edited 10 years ago, but
> Lisa's clock stopped advancing in 1995. In other words, Lisa didn't have a
> year-2000 problem; it had a 1995 problem. What Frank didn't point out was
> that the dialog box gave the elapsed time in years. Most systems in this
> situation would have given a specific last-edit date, or displayed the
> elapsed time as "2555 days". Whoever programmed that dialog box thought to
> convert very-long time intervals into years. What foresight!

Thanks to Larry Tesler. I alway thought it was a great feature.

> 3. During the Lisa demo, Frank Ludolph at one point duplicated a file that
> he meant to move. He was quick to point out that Lisa (and Mac) use the
> same UI for moving and copying documents, and that this was sometimes a
> problem for users, but usually did what users want.

Actually, Mac and Lisa used the same drag-and-drop action, but Mac does a copy when the destination is a different disk while Lisa always did a move. The Mac get beat-up for being inconsistent, but Lisa's consistency lead to difficult to correct errors when an item was dragged from hard disk to diskette - the item was removed from the hard disk and the diskette may have been physically taken to another location.

> ... Behind me in the
> audience, a fellow suddenly said: "Hey! Those two files have the same
> name!" No one else picked up on this, so I told him that in both Lisa and
> Star, filenames have no particular significance to the O.S. and users are
> free to give multiple files the same name if they want to. He found that

> hard to swallow.

I considered mentioning this but cut it from the demo as time was tight. Near the end of the demo I mentioned that the object icons (docs, folders, tools, etc.) were *not* diskfiles and behind an icon could be 0-n diskfiles. (The unique naming being a requirment of the file directory.) This enabled the Lisa to keep two versions of an edited document, 'saved' and 'edits'. The Mac ignored this but NeXT picked it up for their applications and, at 3.0, for other objects as well. This both reduces the number of superflous icons and prevents accidental corruption of the object (move/copy/delete). Removing the 1-1 icon-to-diskfile relationship is one that I'd really like Apple and Microsoft do.

> 4. During the question-answer period, one audience member said: "Seeing
> these two amazing systems makes me wonder what we've been *doing* for the
> past 15 years.   We have color now, what else?" Neither the Star demoers
> nor the Lisa demoers gave him much of an answer. In my opinion, what "we"
> have done over the past 15 years is put these kinds of UIs on the vast
> majority of the huge number of PCs that are in daily use (the "we" being
> mostly Microsoft and Apple, with a little help from unix system vendors).
> The audience member who raised the question was of course thinking more of
> innovation than of proliferation, but in my view the proliferation is a
> more useful and tangible advance than further innovation would have been.

Having flubbed the answer then, I would have answered that we have spent the time, with diminishing returns, on evolving the UI mechanisms. Unfortunately we have been unable to prune off the earlier mechanisms and the result is some rather baroque UIs. I believe that we could build a very good bitmap/mouse UI now if we could do it from the ground up without having to include everyone's favorite feature.

Jeff's comment about proliferation addresses a flubbed answer to a comment from Dave Ungar who said that "these system's really cared about users in a way that today's systems don't seem to." (Or words to that effect.) In those days we were trying to convince people to use computers, that they *could* use computers, in an office environment. Now days, not being able to use a computer is like not having a driver's license in Los Angeles - you can't get by without it. Today users are over a barrel. They have to invest the effort to learn how to use the machine no matter how difficult - it's part of the job requirement. And in the end useability issues rank in priority no higher than 3rd behind features and performance. Just read all the reviews...

When I look back at the technical achievements of Star, well, it is just awesome! It took about 15 years for desktop products to begin to match the overall feature set.

Frank"

For additional Lisa info, see the following web site which has LOTS of stuff (much of which is from David Craig it seems):

http://members.tripod.com/jupiterii/lisacontent/info_lisa.html

This site contains the following information:

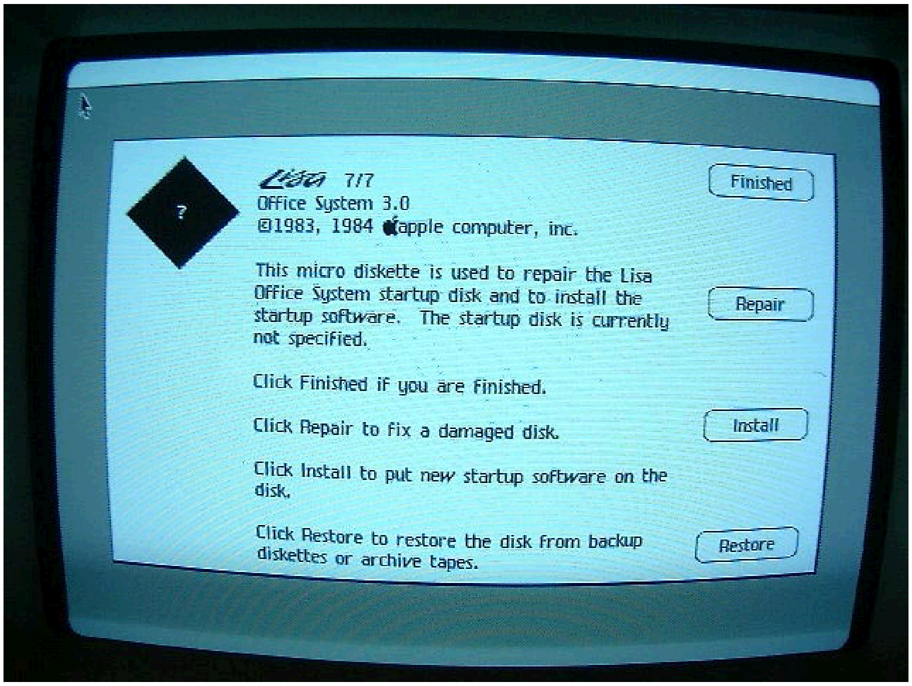
- Lisa Hardware Guide 1981
- Introduction to the Lisa
- Lisa Hardware Guide 1983
- Orphan Support Column From Macazine
- Lisa Schematics
- Lisa Parallel Port Card Docs
- Lisa Product Introduction Plan
- Profile Hard Drive Docs.
- Lisa Marketing Requirements Document
- Lisa Profile Owners Guide
- Lisa Tool Deserialization Paper
- Semaphore Signal Magazine
- Lisa Office System
- Lisa File Label Reader Utility by DTC
- LisaCalc Manual
- LisaTerminal Manual
- Lisa HW Unit Disassembly
- LisaDraw Manual
- Lisa System Low Level Driver Disassembly
- LisaWrite Manual
- LisaGraph Manual
- Lisa OS Reference (1982)
- LisaList Manual
- Lisa OS Guide (1982)
- LisaProject Manual
- Lisa QuickDraw Demo
- Lisa 7/7 Disk Images
- Lisa Pascal 2.0 Reference Manual
- Lisa Pascal Memos
- Lisa SANE slides
- Byte Magazine Article, 9/84
- Lisa Workshop 2.0 Users Guide
- How To Deserialize Lisa Disks
- Lisa Development System Draft 1984
- Press Release: How Lisa Works
- Lisa Independant Developer Opportunities
- Press Release: The Lisa

- Lisa Startup Error Codes
- Lisa to Macintosh Migration Kit
- LisaTalk, Issue #1
- "The Lisa" Article by Edward W. Birss
- Lisa Pascal DevelopmentSystem Manual
- Lisa Pictures
- Lisa 2 Owner's Guide
- Lisa Owner's Guide 1.0
- Lisa Assembly Drawings
- Boot Rom Information
- Lisa Parts List
- General Lisa Information
- Lisa Repair Manual
- Apple-History.com/lisa

(end of message)

Here are a few screen shots of the Lisa showing it in action.

Photo of Installing Lisa Office System 3.0:



Lisa 7/7
Office System 3.0
©1983, 1984 apple computer, inc.

This micro diskette is used to repair the Lisa Office System startup disk and to install the startup software. The startup disk is currently not specified.

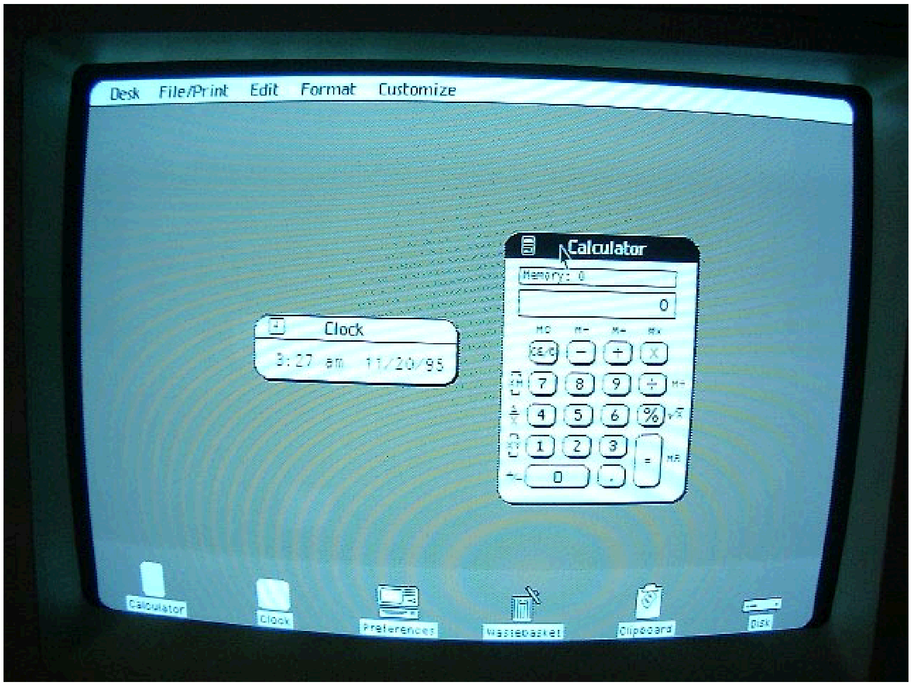
Click Finished if you are finished.

Click Repair to fix a damaged disk.

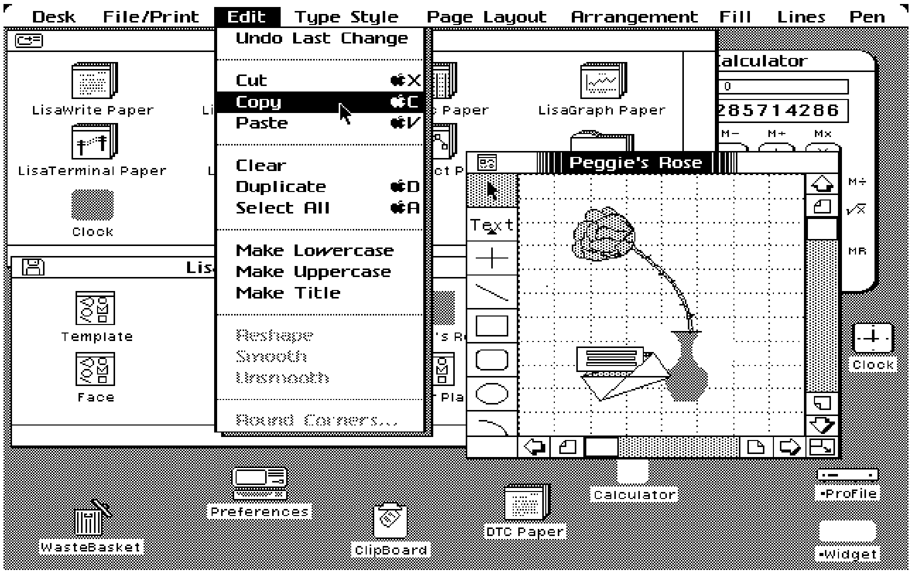
Click Install to put new startup software on the disk.

Click Restore to restore the disk from backup diskettes or archive tapes.

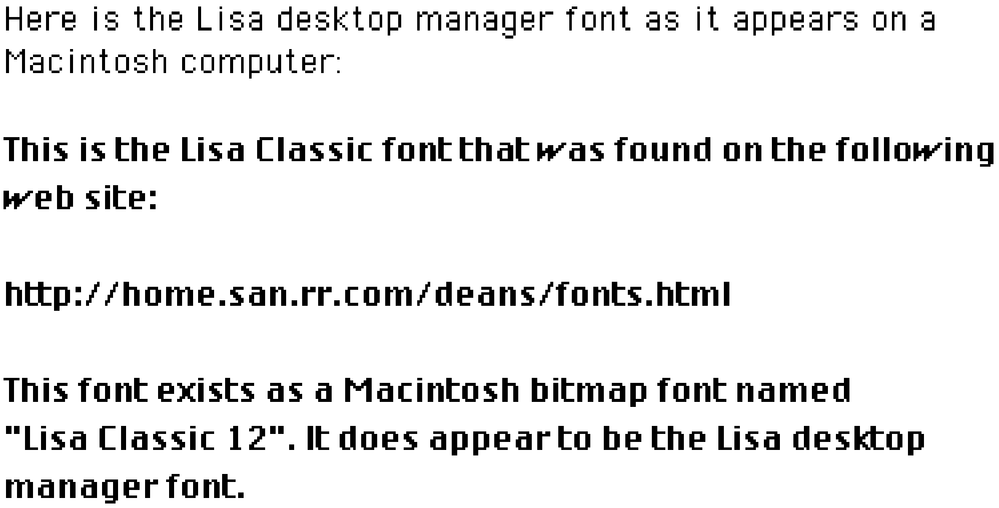
Photo of Lisa Office System 3.0 desktop:



Screen dump of Lisa Office System 3.0 desktop (this is David Craig's desktop, note the "DTC Paper" icon):



Screen shot on Macintosh of Lisa Desktop Manager font:



Here is the Lisa desktop manager font as it appears on a Macintosh computer:

This is the Lisa Classic font that was found on the following web site:

http://home.san.rr.com/deans/fonts.html

This font exists as a Macintosh bitmap font named "Lisa Classic 12". It does appear to be the Lisa desktop manager font.

Regards,
David T. Craig

# David T. Craig -- CyberWolf Inc. -- ACI 4D Developer #5
# Aspen Plaza, 1596 Pacheco, Suite 203
# Santa Fe, NM 87505 USA
# voice 505.983.6463 ext 15 -- fax 505.988.2580
# dcraig@cyberwolf.com