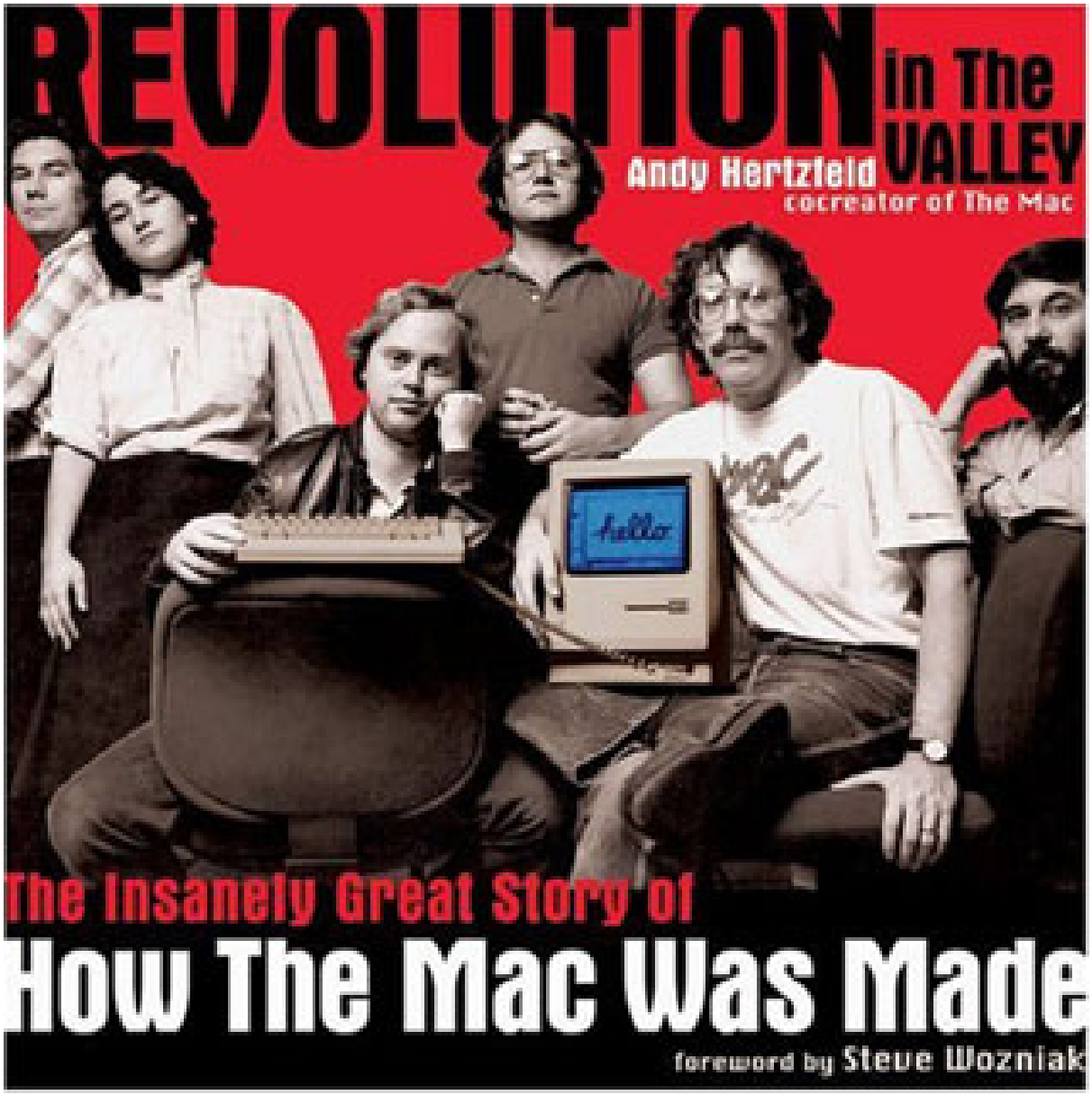

REVOLUTION in The VALLEY
Andy Hertzfeld
cocreator of The Mac
hello
The Insanely Great Story of
How The Mac Was Made
foreword by Steve Wozniak

# I'll Be Your Best Friend

**Author:** Andy Hertzfeld
**Date:** August 1979
**Characters:** Burrell Smith, Andy Hertzfeld, Wendell Sander, Steve Wozniak
**Topics:** Origins, Personality, Hardware Design, Apple II
**Summary:** Burrell Smith was creative in more than just engineering
**Revision:** most recent of 33

**Andy and Burrell in January 1983**

Toward the end of my first week as an Apple employee in August 1979, I noticed that someone had left a black binder on my desk, with a hand-written title that read, "Apple II: Principles of Operation". It contained a brilliant, concise description of how the Apple II hardware worked, reverently explaining details of Woz's epic, creative design hacks, in a clearer fashion than I'd ever read before. I didn't know who left it there, but the title page said it was written by "Burrell C. Smith".

Later that day, in the late afternoon, I was approached by a young, animated, slightly nervous guy with long, straight, blond hair, who entered my cubicle and walked right up to me.

"Are you Andy Hertzfeld? Wow, it's amazing to meet you. I read your articles in Call A.P.P.L.E. and Dr. Dobb's. Apple's lucky they got you to work here. I want to shake your hand."

With exaggerated formality, he extended his right arm stiffly, almost in a parody of a handshake offer. "I'm Burrell. Burrell Carver Smith. Pleased to meet you. I wrote that manual I left on your desk.", he said, pointing to the black binder on my desk. We shook hands and then he suddenly turned around nervously and darted off, without explanation. "See you later!", he said, without looking back.

My cubicle in Bandley 1 was in the hardware engineering section, since my first project was writing the firmware for the Silentype thermal printer. It was across the aisle from Wendell Sander's office. Wendell was the designer of the Apple III and an extremely brilliant and seasoned engineer, who used to design RAM chips for Fairchild and understood the Apple II hardware design inside out. All the other hardware engineers on the team, except for Woz, usually came to Wendell for advice.

I began to notice that Burrell, even though he was supposed to be working in the service department in a different building, often hung around outside of Wendell's office, sometimes for hours at a time, waiting for Wendell to have a free moment, so he could ask him to verify his latest insight about the Apple II timings. Sometimes, when Wendell was busy, he would try the insight out on me instead, or discuss a fine point of the Apple II firmware. Soon, we started to occasionally go out for lunch together.

The first time we went to lunch, I found out that Burrell's creativity extended beyond his engineering work. He would often try to convince our waitress to concoct variations of the standard fare on the menu, thinking of something different every time.

For example, after he successfully persuaded a waitress to divide his pizza toppings into thirds, he asked her to do fifths the next time. Or he would sometimes try to order mixed sodas as if they were cocktails, in ever varying proportions, like three quarters Coke, and one quarter Sprite. Often, the waitress would balk but Burrell was sometimes charming enough to eventually convince her to comply. He would also obsess on certain foods, becoming fixated on Bulgarian Beef sandwiches from Vivi's for a while, and then a Pineapple Pizza phase (see Pineapple Pizza) , evolving to his most enduring favorite, sushi, which provided a new range of interesting choices and combinations.

Burrell also had a distinctive way of expressing himself, applying technical jargon to ordinary life (like a situation that was "meta-stable", or someone being a "state machine"), mixed with a dash of baby talk (like adding plurals to people's names). An attractive woman was referred to as a "good prototype", or a "good proto" for short. Burrell had a great sense of humor and periodically performed hilarious impressions of everyone else on the team, caricaturing their personality quirks with an incisive phrase or nickname (see I Invented Burrell) . He also liked to make fun of various language conventions; for example, if he was pleased with new software, he'd say "Happiness comma software".

One of his favorite expressions was "I'll be your best friend". He offered "best friendship" for a wide range of activities, like making some change in the software for him or getting him a Coke from the gas station. "Best friendship takes place", he would declare if pleased with the results. He also had a habit of reducing things to initials, like "B.F.R." for "Best Friendship Relationship".

Once, right after Burrell conferred best friendship upon me, I heard him offer best friendship to someone else, for a different favor. "Wait a second", I challenged Burrell, "How can you give out best friendship to someone else? There can be only one best friend at a time, can't there?"

Burrell had a quick reply, delivered with a smile. "Of course there can be only one "best" friend at a given instant of time. But best friendship relationships may be highly dynamic. The average length of a best friendship is three to five milliseconds. So there's no problem in having a new B.F.R. a second or two later."

# We'll See About That

**Author:** Andy Hertzfeld
**Date:** November 1979
**Characters:** Burrell Smith, Bill Atkinson, Jef Raskin, Steve Wozniak
**Topics:** Hardware Design, Technical, Apple II
**Summary:** Burrell proves his mettle with the 80k language card.
**Revision:** most recent of 6

**Apple II Language Card**

Burrell Smith was a 23 year old, self taught engineer, without a college degree, who was drawn to Apple by the sheer elegance of the Apple II design. He was hired into Apple in February 1979, as Apple employee #282, a lowly service technician responsible for fixing broken Apple IIs that were sometimes returned by customers. As he debugged broken logic boards, sometimes more than a dozen in a single day, he began to develop a profound respect and empathy for Steve Wozniak's unique, creative design techniques.

Meanwhile, the Lisa team had been writing their first code in Pascal, running on Apple IIs, because the Lisa hardware wasn't ready yet. They had been at it for almost a year, and they had written more code than would fit in the 64 Kbytes of memory in a standard Apple II. In fact, the Apple II only had 48 Kbytes on its main board, but it used a "language" card to give it an extra 16 Kbytes used to run Pascal. To accomplish this, the language card had to "bank switch" its RAM over the ROM on the Apple II motherboard.

Bill Atkinson was the main programmer for both the Apple II Pascal system as well as the new Lisa system. He was in the service department picking up some extra language cards when Burrell heard him lamenting about overflowing the Apple II's memory limitations.

"Well, why don't you add more memory to the language card?", Burrell suggested.

Bill was intrigued, but he complained, "You can't add any more memory because we're out of address space. 64K is the limit of what we can address."

Burrell had already thought of that. "Well, the language card is already bank-switching the RAM, even double-banking the last 2K where the monitor ROM is. We'll just make it bank-switch another bank."

Bill was enthusiatic, so Burrell built him a prototype while Bill modified the Pascal run-time to support the extra bank switching. It worked like a charm, so soon Burrell was busy manufacturing 80K language cards for all the Lisa programmers.

Around this time, Bill ran into Jef Raskin. Jef had been writing a series of papers about a consumer-oriented computer that would be extremely inexpensive and radically easy to use. He was ready to start building a hardware prototype so he was looking for a talented hardware designer who could pull off his vision of a brutally simple, ultra low cost machine.

"I've got someone who you ought to meet", Bill told Jef. He made arrangements to bring Burrell over to Jef's house in Cupertino over the weekend.

Bill and Burrell showed up at Jef's house at the appointed time. Bill introduced Burrell to Jef, saying. "Jef, this is Burrell. He's the guy who's going to design your Macintosh for you."

"We'll see about that", Jef replied. "We'll see about that."

# Scrooge McDuck

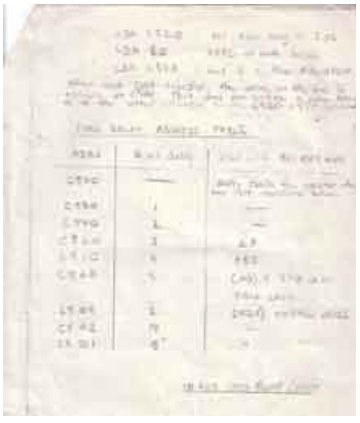
**Author:** Andy Hertzfeld
**Date:** February 1980
**Characters:** Burrell Smith, Andy Hertzfeld, Jef Raskin, Tom Whitney, Bob Bishop, Cliff Huston, Dick Huston
**Topics:** Origins, Hardware Design, Prototypes
**Summary:** The very first image on the very first Macintosh
**Revision:** most recent of 7

**this image is similar to the first image ever displayed on a Macintosh**

Burrell Smith liked to do intensive design work over the Christmas break, so the very first prototype of the very first Macintosh sprung to life early in the first month of the new decade, in January 1980. It wasn't really a stand-alone computer yet, as the prototype resided on an Apple II peripheral card, but it already contained the essential hardware elements of Jef Raskin's Macintosh dream: a Motorola 6809E microprocessor, 64K of memory, and a 256 by 256 bit-mapped graphic frame buffer, which was hooked up to a cute, tiny 7 inch black and white display. Burrell used the Apple II host to poke values into the memory of the prototype, so he could initialize the control registers and run small programs with the 6809.

I went out to lunch with Burrell a few weeks later and, knowing my appreciation for Woz-like hardware hacks, he explained the crazy way that he contrived for the Apple II to talk with the prototype. He didn't want to waste time designing and wiring up hardware to synchronize the memory of the two machines, since that wouldn't be needed by the real product. Instead, he delegated the memory synchronization to the software, requiring the Apple II to hit a special memory address to tell the prototype how many microseconds later to grab data off of the common data bus. It was weird enough to make me interested in seeing if it really worked.

By now, Burrell thought that he had the graphics running properly, but he wasn't really sure; he still needed to write some software to try it out. I told him that I'd look into it when I had some time. He gave me a copy of a handwritten page that contained the magic addresses that I'd have to use, hoping that I'd get around to it soon.



I was used to coming back to the lab at Apple after dinner, to see if anything interesting was going on and working on various extra-curricular projects. I had some spare time that night, so I got out Burrell's instructions and wrote an Apple II (6502) assembly language routine to do the necessary bit-twiddling to transfer whatever was on the Apple II's hi-res graphic display to the Mac prototype's frame-buffer, using Burrell's unusual synchronization scheme. One of my recent side projects involved using Woz's new, one-to-one interleave floppy disk routines to make very fast slideshow disks on the Apple II. I had just made one full of Disney cartoon characters, that were scanned by Bob

Bishop, one of the **Burrell's DMA instructions** early Apple software magicians. Bob adored the work of Carl Barks,
the Disney artist who specialized in Donald Duck, and he had scanned dozens of Barks' Donald Duck images for the Apple II. I selected an image of Scrooge McDuck sitting on top of a huge pile of money bags, blithely playing his fiddle, with a big grin on his beak. I'm not sure why I picked that one, but it seemed to be appropriate for some reason.

Even though it was starting to get late, I was dying to see if my routine was working properly, and it would be very cool to surprise Burrell when he came in the next day with a detailed image on the prototype display. But when I went to try it, I noticed that Burrell's Apple didn't have a disk controller card, so there was no way to load my program. Damn! I couldn't shut the computer down to insert the card, because I didn't know how to reinitialize the Macintosh board after power-up; Burrell hadn't left the magic incantation for that. I thought I was stuck, and would have to wait until Burrell came in tomorrow.

The only other person in the lab that evening was Cliff Huston, who saw the trouble I was having. Cliff was another early Apple employee, who was Dick Huston's (the heroic programmer who wrote the 256-byte Apple II floppy disc boot ROM) older brother and an experienced, somewhat cynical technician. I explained the situation to him and was surprised when he started to smile.

Cliff told me that he could insert a disk controller card into Burrell's Apple II with the power still on, without glitching it out, a feat that I thought was miraculous - you'd have to be incredibly quick and steady not to short-circuit any of the contacts while you were inserting it, running the risk of burning out both the Apple II and the card. But Cliff said he'd done it many times before: all that was required was the confidence that you could actually do it. So I crossed my fingers as he approached Burrell's Apple like a samurai warrior, concentrating for a few seconds before holding his breath and slamming the disk card into the slot with a quick, stacatto thrust.

I could barely make myself look, but amazingly enough Burrell's machine was still running, and the disk booted up so I could load the Scrooge McDuck image and my new conversion routine. And even more suprising, my routine actually worked the first time, displaying a crisp rendition of Uncle Scrooge fiddling away on the Mac's tiny monitor. The Apple II only had 192 scan-lines, while the embryonic Macintosh had 256, so I had some extra room at the bottom where I rendered the message "Hi Burrell!" in a nice-looking twenty-four point, proportional font.

By the time I came in the next morning, an excited Burrell had already showed the image to everyone he could find, but then he accidentally reset the prototype somehow, and didn't know how to get the image back on the screen. I loaded it again so he could show it to Tom Whitney, the engineering VP. I think Jef was pretty pleased to see his new computer start to come alive, but I don't think he was very happy about me giving the demo, since he thought I was too much of a hacker, and I wasn't supposed to be involved with his pet project.

# It's The Moustache That Matters

**Author:** Andy Hertzfeld
**Date:** September 1980
**Characters:** Burrell Smith, Tom Whitney, Steve Wozniak
**Topics:** Management, Personality
**Summary:** Burrell wants to get promoted to engineer
**Revision:** most recent of 5

Burrell was hired into Apple in February 1979 as Apple employee #282, in the lowly position of service technician, one of the lowest paying jobs at the company. Even though he'd been doing genius quality work as a hardware designer on the Macintosh project for a while now (more than nine months), and he was even filling in for Steve Wozniak on the low cost Apple II project, he still wasn't officially promoted to engineer as he requested, which was getting pretty frustrating.

Burrell started thinking about what it would take to get promoted. It obviously wasn't a matter of talent or technical skill, since he was already far more accomplished in that regard than most of the other hardware engineers. It wasn't a matter of working harder, since Burrell already worked harder and was more productive than most of the others. Finally, he noticed something that most of the other engineers had in common that he was lacking: they all had fairly prominent moustaches. And the engineering managers tended to have even bigger moustaches. Tom Whitney, the engineering VP, had the largest moustache of all.

So Burrell immediately started growing his own moustache. It took around a month or so for it to come in fully, but finally he pronounced it complete. And sure enough, that very afternoon, he was called into Tom Whitney's office and told that he was promoted to "member of technical staff" as a full-fledged engineer.

# Good Earth

**Author:** Andy Hertzfeld
**Date:** October 1980
**Characters:** Bud Tribble, Steve Jobs, Jef Raskin, Bill Atkinson, Rod Holt, Burrell Smith, Joanna Hoffman, Brian Howard, Steve Wozniak, Mike Scott, Mike Markkula
**Topics:** Apple Spirit, Origins, Management, Buildings
**Summary:** The original Mac team's original office
**Revision:** most recent of 11

**Nerf balls were a favorite toy**

In 1979 and 1980, Jef Raskin's Macintosh project was a four person research effort with a tenuous existence. It wasn't considered to be very important within Apple, and was almost cancelled a couple of times. When Apple had another major reorganization in the fall of 1980, it was terminated again, but Jef pleaded with Mike Scott and Mike Markkula for more time, and was granted three more months to show that he was really onto something. As part of the re-org, the four person Macintosh team (Jef Raskin, Brian Howard, Burrell Smith, Bud Tribble, soon to be joined by Joanna Hoffman) relocated to a small office building a few blocks from the main Apple campus.

The new office, located at 20863 Stevens Creek Boulevard, was called the "Good Earth" building, because it was adjacent to a Good Earth restaurant. In fact, the office used to be Apple's very first office in Cupertino, after they moved out of Steve Jobs' parents' house, and was later used as the first office of the Lisa project, when the Lisa team had fewer than ten employees. The Mac team moved in, outfitting it with lots of bean bag chairs and all kinds of interesting toys.

Jef was very playful and always encouraged his team to express themselves creatively, so the office quickly began to look more like a day care center than an engineering lab, overflowing with playthings. Periodically, work would cease and the entire team, plus any visitors who might be on the premises, would play some organized game, usually led by Jef and Brian.

The favorite game, which was usually played at least once a day, just after lunch, was a form of tag played with Nerf balls. There were dozens of brightly colored Nerf balls scattered around the office. The rules would be improvised, but usually the person who was "it" had to confer it-hood on someone else by hitting them with a Nerf ball. This inspired everyone to surround their work area with barricades made out of cardboard, to provide cover during the game, making part of the office look like a cardboard maze.

Jef and Brian were both serious musicians, so the office was also littered with a variety of musical instruments, sometimes erupting in spontaneous concerts. Another one of Jef's interests was model airplanes and automobiles, especially radio controlled ones. It wasn't unusual to see a radio controlled car dart underneath your desk, and occasionally everyone would go outside to see the maiden voyage of the latest plane.

Jef was writing his "Book of Macintosh" during much of 1979 and all of 1980, articulating his vision in ever finer detail. Burrell's 6809-based prototype came alive in the early part of 1980, but then he went off to work on the low cost Apple II project. Jef hired Marc Lebrun to write software in early 1980, but Marc was more interested in Lisp machines than a limited memory microcomputer like the Mac, so nothing much happened until he was replaced by Bud Tribble in September 1980.

Bud knew Jef from UCSD, and was also good friends with Bill Atkinson. They had a part time, two person consulting company together in Seattle called Synaptic Systems while they were both graduate students. Bill and Jef convinced Bud to take a one year leave of absence from the M.D/Ph.D. program he was pursuing at University of Washington at Seattle. Bud was in the fifth year of a seven M.D/Ph.D. program. Instead of returning to med school, Bud moved into a spare room at Bill Atkinson's house, and started work on the Mac project at Apple. He quickly began to breathe life into Burrell's languishing prototype, writing some graphics routines for the 6809.

So even though the Mac project had been going for more than a year, the move to Good Earth in October 1980 came at an interesting time, with a new but limited lease on life, and software finally starting to happen. But the Good Earth era was rather short lived.

Around two months after the move, Bud convinced Burrell to consider using the 68000 processor instead of the 6809. Burrell came up with a brilliant design, catching the attention of Steve Jobs. Steve took over the project and quickly recruited most of the early Apple II crew that he trusted, including Steve Wozniak and Rod Holt, and moved the project to larger offices a half mile away, in Texaco Towers (see Texaco Towers) .

# I Invented Burrell

**Author:** Andy Hertzfeld
**Date:** 1981
**Characters:** Burrell Smith, Jef Raskin
**Topics:** Personality
**Summary:** Burrell imitates Jef
**Revision:** most recent of 1

Burrell had a great sense of humor, and he was capable of performing devastating impersonations of everybody else on the Mac team, especially the authority figures.

Whatever idea that you came up with, Jef Raskin had a tendency to claim that he invented it at some earlier point. That trait was the basis of Burrell's impersonation of Jef.

Jef had a slight stammer, which Burrell nailed perfectly. Burrell began by folding his fingers together like Jef and then exclaiming in a soft, Jef-like voice, "Why, why, why, I invented the Macintosh!"

Then Burrell would shift to his radio announcer voice, playing the part of an imaginary interviewer. "No, I thought that Burrell invented the Macintosh", the interviewer would object.

He'd shift back to his Jef voice for the punch line.

"Why, why, why, I invented Burrell!"

# Texaco Towers

**Author:** Andy Hertzfeld

**Date:** January 1981

**Characters:** Steve Jobs, Dan Kottke, Brian Howard, Bud Tribble, Jef Raskin, Burrell Smith, George Crow

**Topics:** Origins, Lisa, Buildings

**Summary:** The office where the Mac became real

**Revision:** most recent of 23

The main Apple buildings on Bandley Drive in Cupertino had boring numerical appellations (Bandley 1, Bandley 3, etc.), but from the beginning the Lisa team gave the buildings they inhabited more interesting names. The original office for the Lisa team was adjacent to a Good Earth restaurant (in fact, it was Apple's original office in Cupertino), so it was called the "Good Earth" building. When the team grew larger and took over two nearby office suites, they were designated "Scorched Earth" (because it housed the hardware engineers, who were all smokers) and "Salt of the Earth".

When the Lisa team became a separate division in 1980, they moved to a larger, two-story office building a block or two away from the main building on Bandley Drive. Everyone was so impressed at having two stories (all the other Apple buildings were single story) that the building was dubbed "Taco Towers", although I'm not sure where the "Taco" part came from.

In December 1980, the embryonic Macintosh team was residing in the Good Earth building, which was abandoned by the Lisa team for Taco Towers earlier in the year. When Steve Jobs took over the project, he moved it to a new building that was large enough to hold about fifteen or twenty people, a few blocks away from the main Apple campus at the southeast corner of Stevens Creek Boulevard and Saratoga-Sunnyvale Road.

There was a Texaco gas station at the corner, and a two-story, small, brown, wood paneled office building behind it, the kind that might house some accountants or insurance agents. Apple rented the top floor, which had four little suites split by a corridor, two on a side. Because of the proximity of the gas station and the perch on the second story, as well as the sonic overlap between "Taco" and "Texaco", the building quickly became known as "Texaco Towers".

Burrell Smith and Brian Howard took over the side of the building closest to the gas station and built a hardware lab, while Bud Tribble and Jef Raskin set up shop on the other side, installing desks with prototype Lisas to use for software development. Bud's office had four desks, but he was the only one occupying it at first. Steve didn't have an office there, but he usually came by to visit in the late afternoon.

In the corner of Bud's office, on one of the empty desks, was Burrell's 68000 based Macintosh prototype, wired-wrapped by Burrell himself, the only one currently in existence, although both

Brian Howard and Dan Kottke had started wire-wrapping additional ones. Bud had written a boot ROM that filled the screen with the word "hello", rendered in a small bitmap that was thirty two pixels wide for easy drawing, which showed off the prototype's razor sharp video and distinctive black on white text.

When I started on the project in February 1981, I was given Jef's old desk in the office next to Bud's. Desk by desk, Texaco Towers began to fill up, as more team members were recruited, like Collette Askeland to lay out the PC boards, or Ed Riddle to work on the keyboard hardware. When George Crow started, there wasn't an office available for him, so he set up a table in the common foyer and began the analog board design there.

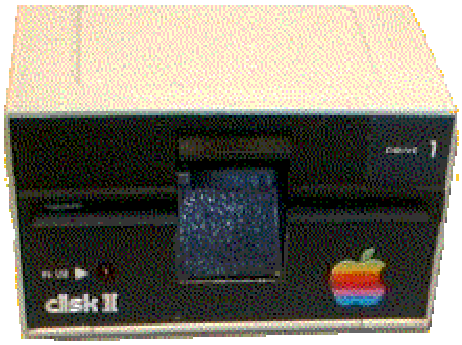
Burell and I liked to have lunch at Cicero's Pizza, which was an old Cupertino restaurant that was just across the street. They had a Defender video game, which we'd play while waiting for our order. We'd also go to Cicero's around 4pm almost every day for another round of Defender playing; Burrell was getting so good he would play for the entire time on a single quarter (see Make a Mess, Clean it Up!) .

In May of 1981, Steve complained that our offices didn't seem lively enough, and gave me permission to buy a portable stereo system for the office at Apple's expense. Burrell and I ran out and bought a silver, cassette-based boom box right away, before he could change his mind. After that we usually played cassette tapes at night or on the weekends when there was nobody around that it would bother.

By early 1982, the Mac team was overflowing Texaco Towers and it was obvious that we'd have to move to larger quarters soon. Steve decided to move the team back to the main Apple campus, into Bandley 4, which had enough space for more than 50 people. The 68000 based Macintosh was born in the Good Earth building, but I still think of Texaco Towers as the place where it came of age, transitioning from a promising research project into a real, world-changing commercial product.

# Nybbles

**Author:** Andy Hertzfeld
**Date:** February 1981
**Characters:** Andy Hertzfeld, Steve Wozniak, Burrell Smith, Brian Howard
**Topics:** Hardware Design, Prototypes, Software Design, Early Programs, Technical
**Summary:** My first job on the Macintosh project
**Revision:** most recent of 7




**An Apple II Disk Drive**

My very first job on the Mac project was to help Burrell Smith and Brian Howard verify that the disk controller was working properly. They had just wired up a Woz-style disk controller to the Macintosh prototype, but they had no way to test if it was working properly without writing a fairly complicated program.

When I arrived on the scene, they were trying to debug a small 68000 routine that they had written together. They had written a simple loop to access the disk controller and were watching it execute with a logic analyzer. But neither of them were much of a programmer, so I was able to help right away, even though I had never programmed the 68000 before.

The Apple II disk controller, designed six months after the Apple II itself was complete, was Steve Wozniak's crowning achievement. His five chip disk controller card out-performed competitive controllers that were four times as expensive by shifting most of the responsibilities from hardware to software. In Woz's approach, the software was responsible for doing all of the encoding and decoding, head stepping, etc. This allowed Woz to improve the capacity and performance over standard techniques.

In those days, most floppy disks used a recording technique called FM encoding, where one clock bit would precede each data bit to improve reliability. But that was overkill, it was possible to get more data on a disk if you used some of the clocks for data. So Woz used a technique called "group encoding" (although Woz called it "nybblizing") to get five bits out of every eight transitions instead of four. He later figured out how to use six bits out of eight (the "16 sector" format vs. the earlier "13 sector" - non-Wozian drives used 10 sectors) improving the density further.

The software stored eight transitions at a time into the disk controller's shift register. Since each transition was 4 microseconds long, each "nybble" of data had to be provided every 32 microseconds. Eventually, Woz figured out how to decode the data on the fly, in between fetching nybbles, so he could achieve the ultimate in performance, one to one interleave. But he never could figure out how to encode on the fly, since the Apple II's microprocessor just wasn't fast enough, and the timing had to be more precise for writing.

But the Macintosh's microprocessor was at least four times faster than the Apple II, more like 10 times if you were manipulating 32 bit values, so writing one-to-one interleave disk routines was simple on the Macintosh. It was fun for me to achieve the holy grail of disk performance without breaking into a sweat. But first there was a big problem to solve.

Woz's disk technology required that the software feed it new data every 32 microseconds exactly. If we were even a single microsecond early or late, it would cause a glitch in the data and ruin it. In order to write the routines, I needed to know how fast the Macintosh executed each instruction.

The manual gave the number of clocks for each instruction, but I wasn't sure how long it took to fetch from memory. So of course, I asked Burrell what the timings were, but I was surprised at his response.

"I don't know. The Mac is synchronous, just like the Apple II, so each instruction has the same timing, every time you execute it, so you will be able to write disk routines that have exact timing. I don't know what it is, so we'll just measure it. Why don't you write your routine and we'll measure it with the logic analyzer."

So I spent a couple of days writing the basic routines, and then sat down with Burrell and Brian in front of the logic analyzer and we watched each instruction execute, writing down how long each instruction took. They usually worked like we expected, but occasionally some things were surprising and I had to adjust the code. After a few fixes, I had the raw disk routines both reading and writing, doing the encoding and the decoding on the fly, achieving Woz's long sought after one-to-one interleave.

I expected to feel elated when I finally got the disk reading and writing, but it didn't feel that satisfying, because you couldn't really see it in action. The previous year, soon after Woz wrote his one-to-one interleave read routines, I made some fast slideshow disks for the Apple II, where the screen was filled with a new image in less than a second, twice as fast as previously possible. I thought it would be fun to use the new disk routines to read the slideshow disks on the Mac.

By this time, it was around 7pm and everyone else was going out to dinner. I was invited to come, but I was so close to getting the slideshow working, that I didn't think that I could concentrate on eating until I got it done. I was alone in the office when I finally got it working, the embryonic Mac reading and displaying images from an Apple II disc as fast as possible. It was far and away the coolest thing a Mac could do so far. It was fun to show it to everyone when they came back from dinner.

# Reality Distortion Field

**Author:** Andy Hertzfeld
**Date:** February 1981
**Characters:** Steve Jobs, Bud Tribble
**Topics:** Management, Personality, Reality Distortion
**Summary:** Bud defines Steve's unique talent
**Revision:** most recent of 6

**Steve Jobs**

I officially started on the Mac project on a Thursday afternoon, and Bud Tribble, my new manager and the only other software person on the project, was out of town. Bud was on leave of absence from an M.D.-Ph.D. program and he had to occasionally return to Seattle to keep up his standing in the program.

Bud usually didn't come into work until after lunch, so I met with him for the first time the following Monday afternoon.
We started talking about all the work that had to be done, which was pretty overwhelming. He showed me the official schedule for developing the software that had us shipping in about ten months, in early January 1982.

"Bud, that's crazy!", I told him. "We've hardly even started yet. There's no way we can get it done by then."

"I know," he responded, in a low voice, almost a whisper.

"You know? If you know the schedule is off-base, why don't you correct it?"

"Well, it's Steve. Steve insists that we're shipping in early 1982, and won't accept answers to the contrary. The best way to describe the situation is a term from Star Trek. Steve has a reality distortion field."

"A what?"

"A reality distortion field. In his presence, reality is malleable. He can convince anyone of practically anything. It wears off when he's not around, but it makes it hard to have realistic schedules. And there's a couple of other things you should know about working with Steve."

"What else?"

"Well, just because he tells you that something is awful or great, it doesn't necessarily mean he'll feel that way tomorrow. You have to low-pass filter his input. And then, he's really funny about ideas. If you tell him a new idea, he'll usually tell you that he thinks it's stupid. But then, if he actually likes it, exactly one week later, he'll come back to you and propose your idea to you, as if he thought of it."

I thought Bud was surely exaggerating, until I observed Steve in action over the next few weeks. The reality distortion field was a confounding melange of a charismatic rhetorical style, an indomitable will, and an eagerness to bend any fact to fit the purpose at hand. If one line of argument failed to persuade, he would deftly switch to another.

Sometimes, he would throw you off balance by suddenly adopting your position as his own, without acknowledging that he ever thought differently.

Amazingly, the reality distortion field seemed to be effective even if you were acutely aware of it, although the effects would fade after Steve departed. We would often discuss potential techniques for grounding it (see Are You Gonna Do It?) , but after a while most of us gave up, accepting it as a force of nature.

# Black Wednesday

**Author:** Andy Hertzfeld
**Date:** February 1981
**Characters:** Andy Hertzfeld, Mike Scott, Steve Jobs, Dick Huston, Donn Denman, Rick Aurrichio, Jef Raskin, Burrell Smith, Brian Howard, Mike Markkula
**Topics:** Origins, Apple II, Management, Recruiting
**Summary:** A shakeup in Apple II engineering frees me up to work on the Macintosh
**Revision:** most recent of 5

I could tell there was something wrong from the moment I stepped into the building, on the morning of Wednesday, February 25th, 1981. Instead of the normal office buzz, there was a muted sadness hanging in the air. People were standing around, huddled in small groups. I ran into Donn Denman, who had a cubicle near mine, and asked him what was going on.

"Didn't you hear? Scotty fired almost half of the Apple II engineering team this morning. He started calling people into his office since around 9am, one at a time, and telling them that they were being fired. I think over thirty people have been fired so far. No one knows why, or who's going to be next. There's going to be a meeting out back around noon when he's supposed to tell us what's going on."

Apple had just gone public a couple of months ago, and it was still growing at a frenzied pace. Sales were booming and there was no financial reason to pare back. I wondered what was going on.

"Do you know who they fired?", I asked Donn.

"Yeah, it's amazing. Scotty fired three out of the four managers, so almost everyone's boss is gone. And believe it or not, they fired Rick Aurrichio."

I thought that the managers were more or less incompetent, so that didn't bother me, but the Rick Aurrichio part was shocking, since Rick was clearly one of the most talented programmers in the Apple II division. He would usually do a week's worth of work in a day or two, and then spend the rest of the week messing around with whatever caught his fancy, usually one of the latest games. I understood how he could be a management challenge, but it made no sense to fire him. He was my partner on the new DOS 4.0 project, which was just getting underway, and the only other programmer besides me that was working on it, so it was especially distressing that they would fire him so abruptly.

So I joined the ranks of the shell-shocked, and listened numbly to the outdoor meeting where Scotty explained his rationale. He said that the company had grown much too fast over the last year or so, and had made a few key bad hires, who themselves had hired even worse people. He thought the Apple II division had become too complacent, and that we had lost the start-up hustle that was the basis of our success. He wanted to shake us out of our complacency and prune out the bad hires, so we could start growing again in the right direction.

Scotty himself seemed a little shaken and unsure. Some of the other senior executives were standing off to the side, but they didn't participate in the meeting.

Woz was there, but didn't look too happy about things. There was a Q&A session at the end of the meeting where a couple employees told Scotty how horribly he handled the situation, but in general everyone seemed listless, as if we didn't know how we should react. Within a few days, everyone was referring to the incident as "Black Wednesday".

Later in the day, I talked to Dick Huston about what had happened. Dick was an early Apple programmer who had written the boot ROM for the disk controller card, who was an astute observer of Apple politics and was friendly with Scotty. He told me that he knew that the purge was going to happen and had even met with Scotty a couple of times in the last week to help him draw up the list of dead weight. He also told me that Scotty had asked for the approval of Mike Markkula and the board of directors, and hadn't received it yet, but decided to go and do it anyway.

I told Dick that I agreed that Apple had made some poor hires over the last year, especially some of the managers, but a Stalin-like purge was not a valid way to run a company. I complained about Rick's firing and told him that the situation made me feel alienated from the company. I was the type of programmer who had to believe in what I was doing, and I wasn't so sure about Apple's values anymore.

When I came in to work the next morning, there was a message on my desk from Mike Scott's secretary, saying that he wanted to talk to me; obviously Dick must have talked to him. I called her back and arranged to show up at his office in an hour. Scotty looked harried, and our conversation was interrupted a few times by various phone calls. Scotty told me that he had heard that I was upset, and thinking about leaving, and wanted me to know that he wanted me to stay. He asked me what he could do to get me excited about Apple again. I told him that I might like to work on the Macintosh, with Burrell and Bud.

Later that afternoon, Scotty's secretary called to tell me that she arranged for me to talk with Steve Jobs. Steve had been involved with the Mac project for more than a month now, and, although I didn't know it at the time, had dismissed the founder of the project, Jef Raskin, the day before, making him take a mandatory leave of absence after Jef had complained about Steve's leadership.

Lots of people at Apple were afraid of Steve Jobs, because of his spontaneous temper tantrums and his proclivity to tell everyone exactly what he thought, which often wasn't very favorable. But he was always nice to me, although sometimes a bit dismissive, in the few interactions that I had with him. I was excited to be talking with him about working on the Mac.

The first thing he said to me when I walked into his office was "Are you any good? We only want really good people working on the Mac, and I'm not sure you're good enough." I told him that yes, I thought that I was pretty good. I was friends with Burrell, and had already helped him out with software a few times.

"I hear that you're creative", Steve continued. "Are you really creative?"

I told him that I wasn't the best judge of that, but that I'd love to work on the Mac, and thought that I'd do a great job. He said he'd get back to me soon about it.

A couple of hours later, around 4:30pm, I was back to work on DOS 4.0 for the Apple II. I was working on low-level code for the system, interrupt handlers and dispatchers, when all of a sudden I notice Steve Jobs peering over the wall of my cubicle.

"I've got good news for you", he told me. "You're working on the Mac team now. Come with me and I'll take you over to your new desk."

"Hey, that's great", I responded. "I just need a day or two to finish up what I'm doing here,

and I can start on the Mac on Monday."

"What are you working on? What's more important than working on the Macintosh?"

"Well, I've just started a new OS for the Apple II, DOS 4.0, and I want to get things in good enough shape so someone else could take it over."

"No, you're just wasting your time with that! Who cares about the Apple II? The Apple II will be dead in a few years. Your OS will be obsolete before it's finished. The Macintosh is the future of Apple, and you're going to start on it now!".

With that, he walked over to my desk, found the power cord to my Apple II, and gave it a sharp tug, pulling it out of the socket, causing my machine to lose power and the code I was working on to vanish. He unplugged my monitor and put it on top of the computer, and then picked both of them up and started walking away. "Come with me. I'm going to take you to your new desk."

We walked outside to Steve's silver Mercedes and he dropped my computer into the trunk. We drove a few blocks to the corner of Stevens Creek and Saratoga-Sunnyvale, to a non-descript, brown-shingled, two story office building next to a Texaco station, while Steve waxed eloquent about how great the Macintosh was going to be. We walked up to the second floor, and into an unlocked door. Steve plopped my system down on a desk in an office near the back of the building and said, "Here's your new desk. Welcome to the Mac team!", before darting off.

I started looking around the office, and saw Burrell Smith and Brian Howard in the next room, huddled over a logic analyzer connected to a prototype board. I told them what happened and they said Steve had been over earlier, asking them if they thought I was any good. They were happy that I joined the team.

After helping them a bit with the disk diagnostic routines they were trying to debug, I returned to my new desk and looked inside the drawers. I was surprised to see that it was still full of someone else's stuff. In fact, the bottom drawer had all kinds of unusual stuff, including various kinds of model airplanes, and some photography equipment. I later found out that Steve had assigned me to Jef Raskin's old desk, which he hadn't had time to move out of yet.

# He's Only in Field Service

**Author:** steve blank
**Date:** March 1981
**Characters:** Burrell Smith, steve blank
**Topics:** Hardware, Hardware Design
**Summary:** The SCC gets designed into the Mac
**Revision:** most recent of 5

In the early eighties, I was at Zilog as the (very junior) product marketing manager for the Z8000 peripheral chips which included the new SCC chip, short for "Serial Communications Controller". I remember getting a call from our local salesman that someone at Apple wanted more technical information than just the spec sheets about our new (not yet shipping) chip. I vividly remember the sales guy saying, "its only some kid in field service, I'm too busy, why don't you drive over there and talk to him."

Zilog was also in Cupertino, near Apple on Bubb drive, and I remember driving to a small non-descript Apple building at the intersection of Stevens Creek and Sunnyvale/Saratoga (most of Apple's buildings at that time were on Bandley Drive.) I had a pleasant meeting and was as convincing as a marketing type could be to a very earnest engineer, mostly promising the moon for a versatile but then very buggy piece of silicon. I remember him thanking me for coming, saying we were the only chip company who cared enough to call on him (little did he know.)

I thought nothing about the meeting until years later. Long gone from Zilog I saw the picture of the Mac team. The field service guy I had pitched the chip to was Burrell Smith. The SCC had been designed into the Mac, and some sales guy who was too busy to take the meeting was probably retired in Maui on the commissions.

# More Like A Porsche

**Author:** Andy Hertzfeld
**Date:** March 1981
**Characters:** Terry Oyama, Jerry Manock, Steve Jobs, Jef Raskin
**Topics:** Industrial Design, Hardware, Prototypes
**Summary:** The design of the Macintosh case
**Revision:** most recent of 14

**The original Macintosh industrial design**

In March 1981, I had been working on the Mac team for only a month. I was used to coming back to the office after dinner and working for a few hours in the evening. Even though many of the early Mac team members usually worked late, and we often went out to dinner together, I was by myself one evening when I returned to Texaco Towers after dinner around 8pm. As soon as I entered the building, I heard loud voices emanating from Bud's office, which was adjacent to mine, apparently engaged in a spirited discussion.

"It's got to be different, different from everything else." I recognized Steve Jobs' voice before I saw him as I passed by the door of Bud's office. He was standing near the doorway, near our only working prototype, conversing with someone who I didn't recognize, that Steve introduced to me as James Ferris, Apple's director of Creative Services. "James is helping me figure out what the Mac should look like," he told me.

The plan of record for the Macintosh industrial design was still the one conceived by Jef Raskin, which chose a horizontally oriented, lunch-box type shape, with the keyboard folding up into the lid of the computer for easy transportability, kind of like the Osborne I, which we weren't aware of at the time. But Steve had a real passion for industrial design, and he never seriously considered following Jef's recommendations.

I went into my office and started to program, working on improving the code that drove the serial link between the Mac and Lisa, at Bud's request. But I couldn't help but overhear the passionate discussion taking place next door between Steve and James Ferris. For some reason, they were talking about cars.

"We need it to have a classic look, that won't go out of style, like the Volkswagen Beetle", I heard Steve tell James.

"No, that's not right.", James replied. "The lines should be voluptuous, like a Ferrari."

"Not a Ferrari, that's not right either", Steve responded, apparently excited by the car comparison. "It should be more like a Porsche!" Not so coincidentally, in those days Steve was driving a Porsche 928.

I thought it was kind of pompous to compare computers with sports cars, even metaphorically. But I was impressed with Steve's passion for elegance in the industrial design and his powers of discrimination continually amazed me as the design took shape.

Steve recruited Jerry Manock to lead the industrial design effort. Jerry was the early Apple employee who had designed the breakthrough plastic case for the Apple II, initially as a contractor before signing up as an employee. For the Macintosh, Jerry recruited a talented designer named Terry Oyama, to do most of the detailed drafting of the actual design. The hard tooling for the plastic case was the component with the longest lead time, so we had to get started right away.

A week or so after the car conversation, Steve and Jerry decided that the Macintosh should defy convention and have a vertical orientation, with the display above the disk drive instead of next to it, in order to minimize desktop footprint, which also dictated a detachable keyboard. That was enough of a direction for Terry to draft a preliminary design and fabricate a painted, plaster model.

We all gathered around for the unveiling of the first model. Steve asked each one of us, in turn, to say what we thought about it. I though it was cute and attractive, looking a lot like an Apple II, but with a distinctive personality all its own. But, after everyone else had their say, Steve cut loose with a torrent of merciless criticism.

"It's way too boxy, it's got to be more curvaceous. The radius of the first chamfer needs to be bigger, and I don't like the size of the bezel. But it's a start."

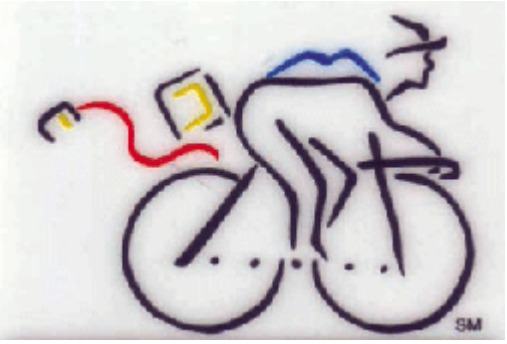
I didn't even know what a chamfer was, but Steve was evidently fluent in the language of industrial design, and extremely demanding about it. Over the next few months, Jerry and Terry iterated on the design. Every month or so, there was a new plaster model. Before a new one was unveiled to the team, Jerry lined up all of the previous ones, so we could compare the new one with past efforts. One notable improvement was the addition of a handle at the top of the case, to make it easier to carry. By the fourth model, I could barely distinguish it from the third one, but Steve was always critical and decisive, saying he loved or hated a detail that I could barely perceive.

At one point, when we were almost finished, Steve called up Jerry over the weekend and told him that we had to change everything. He had seen an elegant new Cuisinart at Macys' on Saturday, and he decided that the Mac should look more like that. So Terry did a whole new design, based around the Cuisinart concept, but it didn't pan out, and soon we were back on the old track, after a one-week diversion.

After five or six models, Steve signed off on the design, and the industrial design team shifted gears to do the laborious engineering work necessary to convert the conceptual model into a real, manufacturable plastic case. In February 1982, it was finally time to release the design for tooling. We held a little party, complete with champagne (see Signing Party) to celebrate sending off the design into the world, the first major component of the Macintosh to be completed.

# Bicycle

**Author:** Andy Hertzfeld
**Date:** April 1981
**Characters:** Rod Holt, Steve Jobs, Mac Team, Jef Raskin
**Topics:** Management, Apple Spirit
**Summary:** Rod wants to change the name of the project
**Revision:** most recent of 6



**Logo for Mac University Consortium**

Jef Raskin chose the name "Macintosh", after his favorite kind of apple, so when Jef was forced to go on an extended leave of absence in February 1981, Steve Jobs and Rod Holt decided to change the name of the project, partially to distance it from Jef. They considered "Macintosh" to be a code name anyway, and didn't want us to get too attached to it.

Apple had recently taken out a two page ad in Scientific American, featuring quotes from Steve Jobs about the wonders of personal computers. The ad explained how humans were not as fast runners as many other species, but a human on a bicycle beat them all. Personal computers were "bicycles for the mind."

A month or so after Jef's departure, Rod Holt announced to the small design team that the new code name for the project was "Bicycle", and that we should change all references to "Macintosh" to "Bicycle". When we objected, thinking "Bicycle" was a silly name, Rod thought that it shouldn't matter, "since it was only a code name".

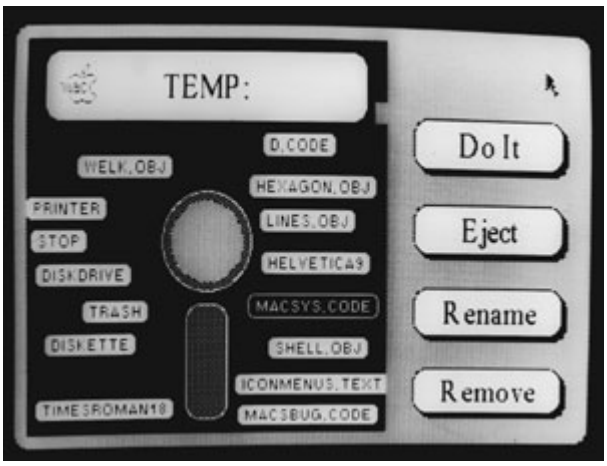
Rod's edict was never obeyed. Somehow, Macintosh just seemed right. It was already ingrained with the team, and the "Bicycle" name seemed forced and inappropriate, so no one but Rod ever called it "Bicycle". For a few weeks, Rod would reprimand anyone who called it "Macintosh" in his presence, but the new name never acquired any momentum. Finally, around a month after his original order, after someone called it "Macintosh" again, he threw up his hands in exasperation and told us, "I give up! You can call it Macintosh if you want. It's only a code name, anyway."

But it was a code name that proved to be sturdy and resilient. In the Fall of 1982, Apple paid tens of thousands of dollars to a marketing consulting firm to come up with a themed set of names for Lisa and Macintosh. They came up with lots of ideas, including calling the Mac the "Apple 40" or the "Apple Allegro". After hearing all the suggestions, Steve and the marketing team decided to go with Lisa and Macintosh as the official names. They did manage to reverse engineer an acronym for Lisa, "Local Integrated Systems Architecture", but internally we preferred the recursive "Lisa: Invented Stupid Acronym", or something like that. Macintosh seemed to be acroynm proof.

But there was still a final hurdle to clear - the name was too close to a trademark from the McIntosh stereo company. I'm not sure how the situation was resolved (I suspect that Apple paid them a modest amount), but toward the end of the retreat in January 1983, Steve announced to the team that we had gotten rights to use the name. He dashed a champagne bottle against one of the prototypes, and declared, "I christen thee Macintosh!"

# Early Demos

**Author:** Andy Hertzfeld
**Date:** April 1981
**Characters:** Andy Hertzfeld, Bud Tribble, Rich Page, Bob Bishop, Bill Atkinson, Bruce Horn
**Topics:** Hardware Design, Prototypes, Software Design. Early Programs
**Summary:** Various demos showed off what the hardware could do
**Revision:** most recent of 10




**Early Finder Prototype from Feb 1982**

he first demo program for the 68000-based Macintosh was written by Bud Tribble, as part of the original boot ROM. It filled the screen with the word 'hello' in tiny letters, more than a hundred times. When the Mac was switched on, it performed some hardware diagnostics, filled the screen with 'hello', and then listened to its serial ports for commands to execute. The 'hellos' told us that everything was working OK.

The boot ROM allowed us to download other programs from the Lisa to the Mac over a serial line, to try out new code and test or demo the prototype. There was a ton of work to do: writing an operating system, hooking up the keyboard and mouse, getting Bill's graphics and UI routines running, and many other tasks, but we also sometimes just did things for fun.

In early March of 1981, I wrote a fast, disk-based slideshow for the Mac the same night that I got the disk routines going (see Nybbles) . It was exciting to see detailed, relatively high resolution images parading across the display so quickly.

By April, I was experimenting with writing custom graphics routines, to show off the raw graphical horsepower of the system. I had written a few ball-bouncing routines on the Apple II, and I thought it would be interesting to see how many balls could the Mac animate smoothly. I wrote some 68000 code to draw 16 by 16 images very quickly, and I found that I could keep more than one hundred balls animating smoothly, which seemed pretty impressive. I also wrote a small sketch program with a seed fill using Bill Atkinson's 8 by 8 pattern bitmaps, as well as an entertaining Breakout game, where I implemented Bud's idea of having to dodge the bricks when they fell down after you hit them.

Bob Bishop had experimented with a variety of graphical special effects on the Apple II, so I thought I'd try some of them out on the Mac. The idea was to transfer an image onto the screen in an entertaining way. The one that I liked best was a kind of waterfall effect, where you copied an image onto the screen using a varying number of multiple copies of successive scan lines, stretching the image vertically. The image looked like it poured onto the screen like water going over a waterfall; it somehow was rather hypnotic. I often used it with an image of the Muppets I converted from the Apple II, and the "Stretching Muppets" demo became pretty well known.

In May 1981, Bud stayed up all night and ported QuickDraw and some pull-down menu code from the Lisa to the Mac (see Busy Being Born, Part 2) . For the first time, we were running mouse-based software with real pull-down menus. The best part of the demo was the pattern menu, which showed off the extensibility of the menu routines to draw an entirely graphical menu.

In June 1981, we realized that it would be worthwhile to create a stand-alone demo environment, where the Macintosh booted and ran programs from its own disk, even though we'd only use it temporarily. Our own operating system wasn't close to usable yet, but Rich Page had written a simple operating system called the "Lisa Monitor" which was based on UCSD Pascal, that was pretty easy to port - all we had to do was integrate our I/O drivers. Soon, using the Monitor, a Mac could boot up and run demos without help from a Lisa.

In the Lisa Monitor environment, it was easy for us to run QuickDraw-based programs. Soon, we had a Window Manager demo, featuring balls bouncing in multiple windows (see Bouncing Pepsis) , as well as a nice icon editor and MacSketch, an early ancestor of MacPaint going.

I think the most interesting early demo was an early prototype for the Finder, written by Bruce Horn and myself in the spring of 1982, and pictured above. Its window was filled with an image of a floppy disc, over which the files were represented as draggable tabs. You could select files and perform operations on them by selecting them and then clicking on command button. Bruce also made a second mock-up, with folder icons, which influenced Bill's design for Lisa's Filer (see Rosing's Rascals) , which we eventually adopted instead. It provides an interesting glimpse of possibilities that we might have chosen instead of what seems so familiar today.

# A Message For Adam

**Author:** Andy Hertzfeld
**Date:** April 1981
**Characters:** Adam Osborne, Steve Jobs, Jef Raskin
**Topics:** Personality, Personality Clashes
**Summary:** We encounter Adam Osborne at the West Coast Computer Faire
**Revision:** most recent of 18

**The Osborne 1 portable computer**

The Apple II was officially introduced at the First West Coast Computer Faire in April 1977, one of the very first trade shows dedicated to the newly emerging microcomputing industry. I loved the Computer Faires because they were attended by passionate hobbyists in the days before commercial forces completely dominated.

In April 1981, a few members of the Mac team took off the afternoon and drove up to San Francisco to visit the seventh West Coast Computer Faire at Brooks Hall. The biggest splash at the show was the unveiling of the Osborne I, from a brand new company named Osborne Computer, which was touted as the world's first portable computer.

The Osborne I was the brain-child of Adam Osborne, who was a well known figure in the world of early microcomputers. Adam was a technical writer who founded a publishing company to publish crucial information about microprocessors and software that was sorely lacking at the time, which was eventually sold to McGraw-Hill. He became a controversial columnist, opining on the industry from his pulpit in InfoWorld and other publications. He had a populist vision of computing, touting a no-frills, low cost, high volume approach to the business.

In 1980, he decided to put his theories into practice, and founded the Osborne Computer Company to design, manufacture and market the Osborne 1, which was a low cost, one-piece, portable computer complete with a suite of bundled applications. He recruited Lee Felsenstein, already a microcomputing legend as the master of ceremonies of the Home Brew Computer Club, to design the hardware. Now, they were introducing the fruits of their labor at the West Coast Computer Faire, as Apple had done four years earlier.

The Osborne 1 was on display at their crowded booth near the center of Brooks Hall. It looked a lot like an oversized lunch box, with a keyboard on the back of the lid, crammed full with two floppy drives and a tiny, 5 inch monitor in its center. We were a little surprised, because it looked uncannily like some of Jef Raskin's early sketches for the Macintosh, which Steve had recently abandoned for a vertically oriented design. Portable was sort of a euphemism as the thing weighed around 25 pounds, but at least it fit under an airline seat, barely. As Macintosh elitists, we were suitably grossed out by the character-based CP/M applications, of course, which seemed especially clumsy on the tiny, scrolling screen.

We worked our way up to the front of the crowd to get a good look at the units that were on display. We started to ask one of the presenters a technical question, when we were suprised to see Adam Osborne himself standing a few feet from us, looking at our show badges, preempting the response.

"Oh, some Apple folks", he addressed us in a condescending tone, "What do you think? The Osborne 1 is going to outsell the Apple II by a factor of 10, don't you think so? What part of Apple do you work in?"

When we told him that we were on the Mac team, he started to chuckle. "The Macintosh, I heard about that. When are we going to get to see it? Well, go back and tell Steve Jobs that the Osborne 1 is going to outsell the Apple II and the Macintosh combined!"

So, after returning to Cupertino later that afternoon, we told Steve about our encounter with Adam Osborne. He smiled, with a sort of mock anger, and immediately grabbed the telephone on the spare desk in Bud's office, and called information for the number of the Osborne Computer Corporation. He dialed the number, but it was answered by a secretary.

"Hi, this is Steve Jobs. I'd like to speak with Adam Osborne."

The secretary informed Steve that Mr. Osborne was not available, and would not be back in the office until tomorrow morning. She asked Steve if he would like to leave a message.

"Yes", Steve replied. He paused for a second. "Here's my message. Tell Adam he's an asshole."

There was a long delay, as the secretary tried to figure out how to respond. Steve continued, "One more thing. I hear that Adam's curious about the Macintosh. Tell him that the Macintosh is so good that he's probably going to buy a few for his children even though it put his company out of business!"

# Square Dots

**Author:** Andy Hertzfeld
**Date:** April 1981
**Characters:** Bill Atkinson, Burrell Smith, Rich Page, Wayne Rosing, Tom Malloy
**Topics:** Hardware Design, Management, Lisa
**Summary:** The Lisa had a different screen resolution than the Macintosh
**Revision:** most recent of 1

**The Apple Lisa**

From the very beginning, even before it had a mouse, the Lisa was designed to be an office machine, and word processing was considered to be its most important application. In the late seventies, the acid test for an office computer (as compared with a hobby computer) was the ability to display 80 columns of text.

The Lisa team decided to optimize their display for horizontal resolution, in order to be able to display 80 columns of text in an attractive font. The vertical resolution wasn't as important, because vertical scrolling works much better for text than horizontal scrolling. The designers decided to endow Lisa with twice as much horizontal resolution as vertical, using a 720 by 360 pixel display, with pixels that were twice as high as they were wide. This was great for text oriented applications like the word processor, but it made things somewhat awkward for the more graphical applications.

When Burrell redesigned the Macintosh in December 1980 to use the same microprocessor as Lisa, the Motorola 68000, it set off shock waves within Apple. Not only was Burrell's new design much simpler than Lisa, with less than half the chip count, but it also ran almost twice as fast, using an 8 megahertz clock instead of a 5 megahertz clock. Among other advantages was the fact that the Mac's 384 by 256 pixel display had the identical horizontal and vertical resolution, a feature that we called "square dots". Square dots made it easier to write graphical applications, since you didn't have to worry about the resolution disparity.

Bill Atkinson, the author of Quickdraw and the main Lisa graphics programmer, was a strong advocate of square dots, but not everyone on the Lisa team felt the same way. Tom Malloy, who was Apple's first hire from Xerox PARC and the principal author of the Lisa word processor, thought that it was better to have the increased horizontal resolution. But Burrell's redesign moved the debate from the theoretical to the pragmatic, by creating a square dots machine to compare with the Lisa.

The Lisa hardware was scheduled to go through a final round of design tweaks, and Bill tried to convince them to switch to square dots. He mentioned his desire to Burrell, who responded by working over the weekend to sketch out a scaled up version of the Macintosh design that featured a full 16-bit memory bus with a 768 by 512 display and square dots, that would also run twice as fast as the current Lisa design. Bill convinced the Lisa engineering manager, Wayne Rosing, that he should at least consider adopting some of Burrell's ideas, and arranged for the leadership of the Lisa team to get a demo of the current Macintosh, and learn about Burrell's new scaled-up design.

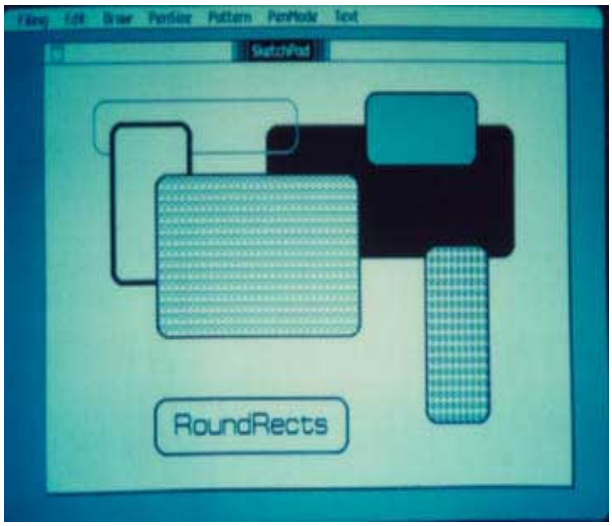
Wayne Rosing led a delegation of his top hardware and software guys over to Texaco Towers for a demo on a Monday afternoon, including hardware guys Rich Page and Paul Baker and software manager Bruce Daniels. We ran various graphics demos, with Bill Atkinson doing the talking, and then Burrell gave a presentation on the Mac design, and his ideas for scaling it up to 768 by 512 display. Everyone seemed pretty impressed and Bill was optimistic that they would make the change.

After a few days, Bill told us the disappointing news that Wayne had decided that there wasn't enough time to embark on such a radical redesign, since at that point the Lisa was scheduled to ship in less than a year. It ended up shipping around two years later with the original 720 by 360 resolution and relatively slow microprocessor, which became a problem when Apple decided to offer a Macintosh compatibility mode for Lisa in 1984. The emulation software didn't try to compensate for the different resolutions, so the applications were distorted by the resolution disparity, almost like looking at a fun-house mirror. It would have continued to be a problem for Apple if the Lisa hadn't been discontinued in 1985.

# Round Rects Are Everywhere!

**Author:** Andy Hertzfeld
**Date:** May 1981
**Characters:** Steve Jobs, Bill Atkinson
**Topics:** Software Design, Inspiration, QuickDraw
**Summary:** Steve inspires Bill by pointing out something about the real world
**Revision:** most recent of 8

Bill Atkinson worked mostly at home, but whenever he made significant progress he rushed in to Apple to show it off to anyone who would appreciate it. This time, he visited the Macintosh offices at Texaco Towers to show off his brand new oval routines, which were implemented using a really clever algorithm.




Bill had added new code to QuickDraw (which was still called LisaGraf at this point) to draw circles and ovals very quickly. That was a bit hard to do on the Macintosh, since the math for circles usually involved taking square roots, and the 68000 processor in the Lisa and Macintosh didn't support floating point operations. But Bill had come up with a clever way to do the circle calculation that only used addition and subtraction, not even multiplication or division, which the 68000 could do, but was kind of slow at.

Bill's technique used the fact the sum of a sequence of odd numbers is always the next perfect square (For example, 1 + 3 = 4, 1 + 3 + 5 = 9, 1 + 3 + 5 + 7 = 16, etc). So he could figure out when to bump the dependent coordinate value by iterating in a loop until a threshold was exceeded. This allowed QuickDraw to draw ovals very quickly.

Bill fired up his demo and it quickly filled the Lisa screen with randomly-sized ovals, faster than you thought was possible. But something was bothering Steve Jobs. "Well, circles and ovals are good, but how about drawing rectangles with rounded corners? Can we do that now, too?"

"No, there's no way to do that. In fact it would be really hard to do, and I don't think we really need it". I think Bill was a little miffed that Steve wasn't raving over the fast ovals and still wanted more.

Steve suddenly got more intense. "Rectangles with rounded corners are everywhere! Just look around this room!". And sure enough, there were lots of them, like the whiteboard and some of the desks and tables. Then he pointed out the window. "And look outside, there's even more, practically everywhere you look!". He even persuaded Bill to take a quick walk around the block with him, pointing out every rectangle with rounded corners that he could find.

When Steve and Bill passed a no-parking sign with rounded corners, it did the trick. "OK, I give up", Bill pleaded. "I'll see if it's as hard as I thought." He went back home to work on it.

Bill returned to Texaco Towers the following afternoon, with a big smile on his face. His demo was now drawing rectangles with beautifully rounded corners blisteringly fast, almost at the speed of plain rectangles. When he added the code to LisaGraf, he named the new primitive "RoundRects". Over the next few months, roundrects worked their way into various parts of the user interface, and soon became indispensable.

# Pineapple Pizza

**Author:** Andy Hertzfeld
**Date:** May 1981
**Characters:** Burrell Smith, Brian Howard, Dan Kottke, Steve Jobs, Bud Tribble
**Topics:** Hardware, Prototypes, Management
**Summary:** We stay late to bring up the first printed circuit board
**Revision:** most recent of 14

**Our reward**

When I began working on the Mac project in February 1981, there was still only one single 68000-based Macintosh prototype in existence, the initial digital board that was wire-wrapped by Burrell himself. It was now sitting in the corner of Bud Tribble's office, on one of the empty desks, attached to a small, seven-inch monitor. When powered up, the code in the boot ROM filled the screen with the word 'hello', in lower case letters and a tiny font, rendered crisply on the distinctive black-on-white display.

Dan Kottke and Brian Howard were already busy wire-wrapping more prototype boards, carefully following Burrell's drawings. In a week or so, I received the second prototype for my office, so I could work on the low level I/O routines, interfacing the disk and keyboard, while Bud worked on the mouse driver and porting Bill's graphics routines.

The next big step for the hardware was to lay out a printed circuit board. We recruited Collette Askeland, the best PC board layout technician in the company, from the Apple II group. Burrell spent a week or two working intensely with Collette, who used a specialized CAD machine located in Bandley 3 to input the topology and route the signals, eventually outputting a tape containing all the information needed to fabricate the boards.

Burrell and Brian Howard checked and rechecked the layout, which was tediously expressed as thousands of node connections, and after a day or two they decided they were ready to send it out for fabrication. We were hoping to get the first sample boards back before the weekend, but it looked like they weren't going to make it. Finally, around 4:30pm on a Friday afternoon, they arrived.

Burrell figured that it would take at least two or three hours to assemble a board, and then even longer to troubleshoot the inevitable mistakes, so it was too late to try to get one working that evening. Maybe they would come in on Saturday to get started, or maybe they'd wait until Monday morning. While they were discussing it, Steve Jobs strolled into the hardware lab, excited as usual.

"Hey, I heard that the PC boards finally arrived. Are they going to work? When will you have one working?"

Burrell explained that the boards had just arrived, and that it would take at least a couple of hours to assemble one, so they were thinking about whether to start tomorrow morning or wait until Monday.

"Monday? Are you kidding?", replied Steve, "It's your PC board, Burrell, don't you want to see if it works tonight? I'll tell you what, if you can get it to work this evening, I'll take you and anyone else who sticks around out for Pineapple Pizza."

Steve knew that Pineapple Pizzas had recently replaced Bulgarian Beef as Burrell's current food obsession (which, as a staunch vegetarian, he thought was a positive development) and that Burrell wanted a Pineapple Pizza pretty much every chance he could get. Burrell looked at Brian Howard and shrugged. "OK, we may as well give it a shot now. But I don't think we'll be able to get it working before the restaurants close."

So Burrell and Brian got busy, selecting a board and stuffing it with sockets, carefully soldering them in place, while five or six of the rest of us, including Steve, sat around and kibitzed. Burrell seemed a little tense and impatient, since he didn't like the pressure of bringing up a board in front of so many spectators. Every five minutes or so, he referred to the awaiting Pineapple Pizza, speculating about how good it was going to taste.

Finally, around 8pm or so, the board was assembled enough to try to power it on for the very first time. The prototype was hooked up to an Apple II power supply and a small monitor, and fired up as we held our breath. The screen should have been filled with 'hellos', but instead all that was there was a checkerboard pattern.

We were all disappointed, except for Burrell. "That's not too bad", he commented, "It means the RAM and the video generation are more or less working. The processor isn't resetting, but it looks like we're pretty close." He turned to look directly at Steve. "But I'm too hungry to keep working - I think it's time for some Pineapple Pizza."

Steve smiled and agreed that it was good enough for the first night, and it was time to celebrate. The seven or eight of us who stayed late drove in three cars to Burrell's favorite Italian restaurant, Frankie, Johnny and Luigi's in Mountain View, ordering three large Pineapple Pizzas, which tasted great.

# Macintosh Prototypes

**Author:** Daniel Kottke

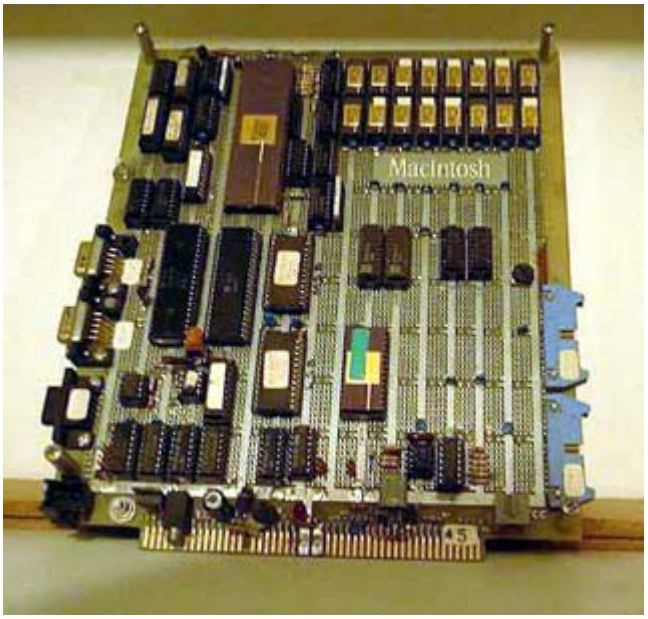
**Date:** June 1981

**Characters:** Dan Kottke, Burrell Smith, Brian Howard, Folon, Jerry Manock, George Crow, Steve Jobs, Wendell Sander, Colette Askeland

**Topics:** Hardware Design, Prototypes, Technical

**Summary:** The evolution of the Macintosh digital board

**Revision:** most recent of 8




**Macintosh wire-wrap #5**

When I started with the Mac team in Jan ’81 there was just one wire-wrapped 68000 prototype, so my first job was to start making more protos along with Brian (‘Uncle Dougie’) Howard – who technically was a writer working under Jef but had been handling many of the hardware build-and-debug tasks for Burrell. Between the two of us Brian and I built another 4 or 5 prototypes for use by the programmers. These were built using some generic 8” x 9” printed circuit cards we found. One of these wire-wrap protos could be built in a couple of days if you had all the parts on hand, and they had the advantage of being readily modifiable. Actually, these protos were undergoing fairly continual modification and the one I that I held on to, #5, had been updated in 1982 with the Z8530 SCC (Serial Comm Controller), the IWM (Integrated Woz Machine) disk controller, and the STF (Special Task Force) clock, which were improvements over the original design.

The basic Mac architecture was very spare, (**„TIGHT!”** as the teenagers now say): about 32 ICs not counting the 68000 and the 16 RAM chips. There was the Timing State Machine (TSM), made out of a PAL and some flipflops; the Linear Address Generator (LAG) made out of a PAL and a couple of counters; the Bus Mgmt Unit (BMU) PAL, 4 multiplexors for the RAM addressing, 2 EPROMs, some bus drivers, a video output shift register, and the 6522 Peripheral Interface Adapter (PIA) which had 16 programmable I/O lines, which handled the keyboard and mouse interfaces and some other tasks. That was the core architecture, to which was added the dual serial ports, the internal and external floppy drive ports, the real-time clock chip, and the sound output. The PALs were the Programmable Array Logic parts made by MMI, which allowed one to write output logic equations to define each of the 8 outputs. They were fairly power-hungry (by today’s standards at least) but were cheap and flexible and Burrell put them to good use.

The underside of the wire-wrap protos was of course a maze of wires. One morning we were amazed to find that one of the protos was badly scorched... it turned out that late the night before, Burrell had been making some changes and managed to get power and ground shorted together somehow, and in his impatience to get the proto back up, he decided to use ‘brute force’ by removing all the chips and connecting 120VAC across the power and ground in an attempt to ‘burn out’ the short.... a misguided attempt, as one would expect! I don’t that proto was ever revived.

Later in 1981 we brought Colette Askeland over from the Apple-II/III division to lay out a Mac PCB – the first of eight iterations of the board between 1981 and the final version in late 1983. Looking at these boards now, they seem like quaint antiques indeed. Not one surface mount component! Here's a list of the 8 board iterations:

**#1 First PCB 1981 MC1001-00** no silkscreen... big holes around the perimeter for mounting the 'fence', the solid copper bus bar that was to run around the perimeter of the board and be tied to cold or chassis ground in an attempt to minimize the RF energy radiation from the digital logic. We eventually were able to cut the 'fence' back to just run along the back edge of the board.

**#2 First Build, 1981 MC1001-01**
--33 ICs (not counting CPU, 128k RAM). 3 Eproms, dual 6551 UARTs (serial ports), 6522 PIA
3 PALs: BMU (Bus Mgmt Unit), LAG (Linear Address Gen.), TSM (Timing State Machine)

**#3 MC1001-02 Feb1982 build of 50**
--32 ICs (not counting CPU, RAM). Z8530 SCC, word-wide RAM (instead of byte-wide), and only 2 EPROM sockets now. At the specific request of our esthetic guru Steve Jobs, the RAM array was spread out to match the regular grid spacing of the other chips, which didn't work out so well: the increased trace lengths caused a troublesome increase in noise on the high-speed RAM signals, so the next board went back to the more densely packed array. This board also included the new STFCLK ('Special Task Force' clock chip) which was a custom chip, dedicated to keeping track of real time and able to store some system preference values. The mouse connector was moved to the back of the board. The Z8530 was a very new part from Zilog which incorporated dual serial ports with a new high-speed synchronous serial mode AND a hardware header-recognition compare register, so that incoming packets could be qualified by the hardware and the processor would only be interrupted to service packets addressed to that Mac -- this was to be key to implementing Appletalk (after a few more years of software development). The Z8530 was a cool new high-performance chip which, like the 68000, seemed over-powered for a small consumer-oriented machine – but Steve Jobs once again successfully negotiated a remarkably low price for the part based on our projections of buying millions of them. This board also marked a change from the old Apple-II 6-pin power connector to a 10 pin power/video connector, no doubt because we finally had George Crow on the team and his 'analog board' integrated the switching power supply and the video CRT electronics. We didn't have the IWM floppy controller ICs yet so this board still uses the old Apple-II floppy disk controller circuit (a state machine made of a prom and a latch).

**#4 MC1019-00 IBM board, summer 1982**
--The 'IBM' (Integrated Burrell Machine) was the custom 64 pin VLSI chip, (same size as the 68000 uP), which integrated a great deal of the core parts of the Mac, in an effort to make the design more cost-effective and resistant to reverse engineering. Unfortunately there were 'low level' problems with the design and the part never worked properly before the project was dropped as being too expensive and risky. This board has the IWM (Integrated Woz Machine), the custom floppy controller chip that Wendell Sander had designed as an improvement over Woz's original Apple-II floppy disk controller.

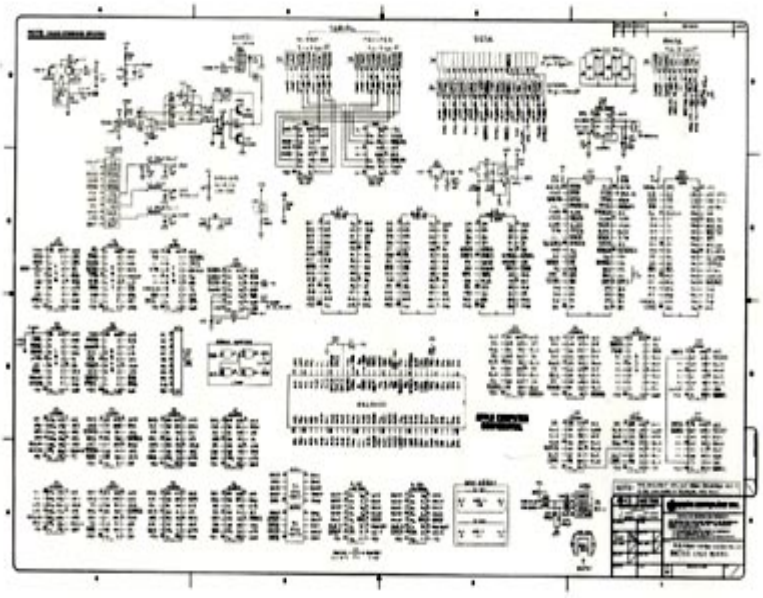
**#5 MC1023-00 MAC512 board, 1982** Now supporting 512k of RAM (still only 16 chips).
--Close to the final version, this board now has 5 PALs: the TSM, LAG, BMU0 and BMU1, and the ASG (Analog Sound Generator). It also has a 4-pin connector (J8) that I recall we added to enable bringing out a few key video signals to allow for creating color or grey-scale video in the future.

**#6 820-0086-00 MAC512, 200 build, May 1983** Folon ‘MacMan’ logo
--Now a 6th PAL has been added, the TSG (Timing Signal Generator).

**#7 820-0086-B** The ‘MacMan’ logo has been dropped, along with the J8 external video connector.

**#8 820-0086-C** final board, 1983 It’s not clear what changes were made for this last rev except a few components appear to have been moved to make room for a ‘Macintosh’ logo along the rear left side.

**Macintosh Schematics**



**Macintosh schematic**

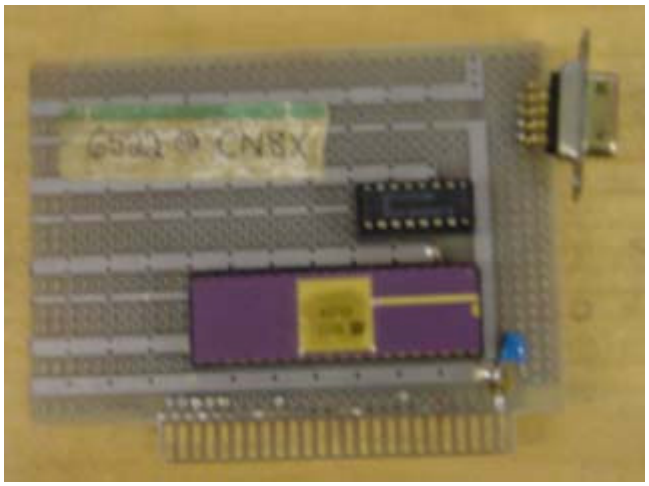
The Macintosh schematics were drawn by Dave Root (I think) in the CAD group, under the direction of lead mechanical designer Jerry Manock. Following industry practice, the set of official Mac schematics comprised something like 3 or 4 large ‘D’ size drawings, with many signals crossing from one page to the other and back again. As the Mac electrical design became frozen in early 1983 I was thinking we needed a simplified condensed schematic, a one-page version that could be reduced and hung on the wall when troubleshooting the logic board. Since so much of the Mac architecture was incorporated in PALs and every input and output of a PAL has a name, it occurred to me to make a new kind of hybrid schematic, where the signal paths were not actually drawn (except in a few cases), but there’s a name for every input and output of every chip on the board. This would be a hybrid of the assembly drawing, which shows every part in its actual location and orientation on the board, and the schematic, which shows what connects to what.

The problem with traditional schematics is that what you usually want to know is where to find a particular signal to look at it, so what you really need is an easy way to find the pin number and where on the chip it is – something the traditional schematics are not that helpful for since the pins on each chip are usually drawn in a function-oriented (or else random) way, rather than drawing the pinout as it actually appears when you look at the part. Imagine a stylized assembly drawing, where each pin of each part has a named signal – that’s the schematic I made. Along with a netlist which listed what each pin connects to, of course – but usually you’d pretty much have that memorized if you were familiar with the architecture.

The Mac CAD Dept didn’t have any use for this kind of hybrid document, and asked me to put a note on it to the effect that it wasn’t a released document but a technical aid only. That was fine with me, I just found it really satisfying to be able to work on diagnosing board problems with just that one piece of 11 x 17 paper (reduced from one ‘D’ page) instead of fumbling around with multiple D-size drawings. Nowadays schematics ARE alot more like that, except still they’re usually spread over many 8 _ x 11” pages and still don’t give any help w/r to finding pins on large surface-mount chips.

# Apple II Mouse Card

**Author:** Andy Hertzfeld
**Date:** June 1981
**Characters:** Andy Hertzfeld, Burrell Smith, Dan Kottke, Bill Budge, Steve Jobs, Bud Tribble, Bill Atkinson
**Topics:** Software Design, Apple Spirit, Prototypes, Technical, Apple II, Compensation
**Summary:** We hooked up a mouse to the Apple II
**Revision:** most recent of 4



**Apple II Mouse Card Protoype**

Burrell's Macintosh design was elegant and clever in many different ways, always using minimal hardware resources for maximal value. The mouse interface was a typical example, using just a fraction of the general purpose 6522 VIA chip that Burrell had incorporated. The mouse is hooked up to the 6522 so that it generates an interrupt each time the mouse moved a notch horizontally or vertically, with a one bit line to sense the mouse button. That was it - the rest was done in software.

Meanwhile, my good friend Bill Budge, who was an amazing Apple II games programmer, came up with an incredibly fast way to draw proportional text on the Apple II graphics screen, using seven pre-shifted tables. He also wrote a very fast BitBlit routine (BitBlit was the most important graphics primitive in GUIs). It was crying out for a way to interface the mouse, so you could have a complete graphical user interface on the Apple II.

Bud Tribble had written the initial code to drive the mouse on the Macintosh, but I had to maintain it, so I was familiar with how it worked. I had used a 6522 chip the previous year, since the Apple III had one, which I had programmed to shift bits out to the Silentype thermal printer. I realized that all that was required to interface a mouse to the Apple III was to make a simple connector to route the appropriate signals to the proper pins.

I asked Dan Kottke to wire up a connector. Once that was finished, it only took a few hours to write cursor routines and a driver. We had the beginnings of a full graphical user interface going on the Apple III, with no additional hardware required.

It was exciting enough that I started helping Bill Budge flesh out the program, careful to work on it only after regular hours. Pretty soon, we had pull-down menus going, and then a simple proportional text word processor. The only problem was that Apple IIIs weren't selling very well; we really needed to get it running on the Apple II somehow.

I talked to Burrell about the problem, which basically came down to making a peripheral card for the Apple II that included a 6522 VIA chip on it. Burrell came back the next day with a brilliant two chip design, just the VIA and a tiny dual flip-flop chip to synchronize it.

To move the mouse cursor without flickering, both the Mac and the Apple III code used a vertical blanking interrupt, which drew the cursor during the time interval when no video was being painted. Unlike the Mac and Apple III, the Apple II didn't have a vertical blanking interrupt. The most brilliant part of Burrell's design involved how it synchronized with the video, without running a wire to pick up the video signal.

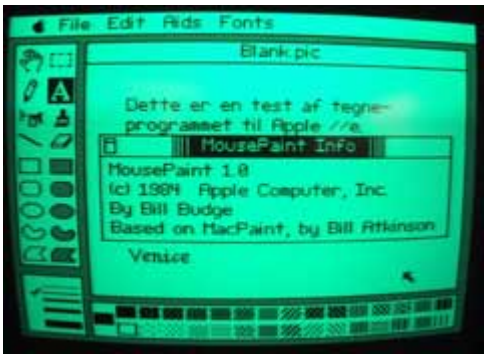
The 6522 chip had a timer which could generate an interrupt at a specified interval. The problem was synchronizing it with the video, since the video generation was not accessible to the processor. Burrell solved the problem by wiring up the spare flip flop to the low bit of the data bus, using it to latch whatever data the video was displaying so the processor could read it.

To synchronize with the video, Burrell had me fill the Apple II's frame buffer so the low bit was on most of the time, but set off at the end of the last scan line. I wrote a routine to sit in a tight loop, reading the latch. When the low bit changed, we would know the vertical blanking interval had just begun.

The Apple II wasn't fast enough to keep up with its own frame buffer. A new byte of video data flew by every microsecond, which was only one processor cycle. It took at least 10 processor cycles to iterate through a loop, so we ran the risk of never seeing the low bit change. Burrell had a way around this, though - if the loop time was relatively prime to the display frequency, it eventually had to slip into place. I wrote a 17 microsecond loop that fit the bill, and we were delighted to see it work perfectly.

Bill Atkinson had told Steve that you couldn't pull off a decent graphical user interface on the Apple II because of the weakness of its 6502 processor and the complexity of Woz's convoluted frame buffer. But, after a little more work on the software, we had a cool little GUI for the Apple II, with a proportional text word processor, in the summer of 1981. We thought that we had had a potentially valuable product on our hands.

We were reluctant to show it to Steve, knowing that he would want to commandeer it, but he heard about it from someone and demanded to see it. We showed it to him, and, unfortunately, he loved it. But he also insisted that Apple owned all the rights to it, even though we had developed it in our spare time.



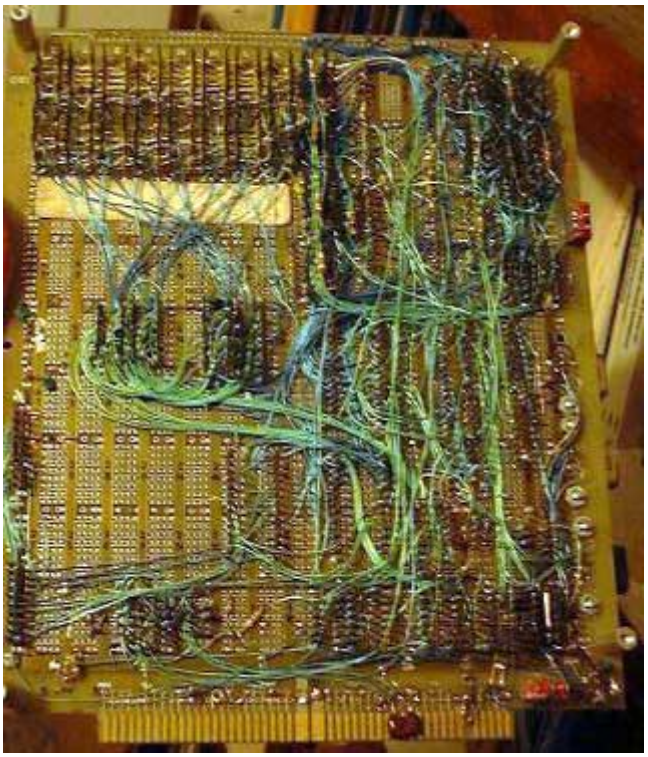

**Bill Budge's MousePaint**

Steve couldn't insist that Apple owned all of it, because Bill Budge wasn't an Apple employee at the time. But Steve could claim complete ownership of the interface card, which he said was developed with Apple resources. Burrell and I were pretty upset, because we did it on our own time and thought that we should be compensated, but it's really hard to argue with Steve, especially about money.

We ended up turning over Burrell's design to the Apple II division, but they didn't think the Apple II could deal with interrupts properly (even though we had demonstrated that it could), so they added tons of hardware and ended up with more than a dozen chips. Steve made a deal with Bill Budge that eventually resulted in MousePaint, a MacPaint clone for the Apple II which was bundled with the mouse card. Burrell and I got over it quickly, but a bit of bitterness lingered, and the whole episode whetted our appetites for eventually working on our own.

# PC Board Esthetics

**Author:** Andy Hertzfeld
**Date:** July 1981
**Characters:** Steve Jobs, Burrell Smith, George Crow, Brian Howard, Dan Kottke
**Topics:** Hardware Design, Personality
**Summary:** Steve is concerned with the esthetics of the PC board
**Revision:** most recent of 4



**A wire-wrapped Macintosh prototype**

The first Mac prototypes were hand-made using a technique called "wire-wrapping", where each individual signal is routed by wrapping an individual wire around two pins. Burrell wire-wrapped the first prototype himself, and then others were done by Brian Howard and Dan Kottke. But wire-wrapping is time consuming and error prone.

By the spring of 1981, the Mac's hardware design was stable enough for us to make a printed circuit board, which would allow us to make prototypes much more quickly. We recruited Collete Askeland from the Apple II group to lay out the board, and after working with Burrell and Brian for a couple of weeks, she taped out the design and sent it off for a limited production run of a few dozen boards.

We started having weekly management meetings in June 1981, which were attended by most of the team, where we discussed the issues of the week. At the second or third meeting, Burrell presented an intricate blueprint of the PC board layout, which had already been used to build a few working prototypes, blown up to four times the actual size.

Steve started critiquing the layout on a purely esthetic basis. "That part's really pretty", he proclaimed. "But look at the memory chips. That's ugly. The lines are too close together".

George Crow, our recently hired analog engineer, interrupted Steve. "Who cares what the PC board looks like? The only thing that's important is how well that it works. Nobody is going to see the PC board."

Steve responded strongly. "I'm gonna see it! I want it to be as beautiful as possible, even if it's inside the box. A great carpenter isn't going to use lousy wood for the back of a cabinet, even though nobody's going to see it."

George started to argue with Steve, since he wasn't on the team long enough to know that it was a losing battle. Fortunately, Burrell interrupted him.

"Well, that was a difficult part to layout because of the memory bus.", Burrell responded. "If we change it, it might not work as well electrically".

"OK, I'll tell you what," said Steve. "Let's do another layout to make the board prettier, but if it doesn't work as well, we'll change it back."

So we invested another $5,000 or so to make a few boards with a new layout that routed the memory bus in a Steve-approved fashion. But sure enough, the new boards didn't work properly, as Burrell had predicted, so we switched back to the old design for the next run of prototypes.

# Diagnostic Port

**Author:** Andy Hertzfeld
**Date:** July 1981
**Characters:** Steve Jobs, Jef Raskin, Burrell Smith, Brian Howard, Steve Wozniak, Rod Holt
**Topics:** Hardware Design, Management
**Summary:** Burrell tried to sneak in some hardware expandability
**Revision:** most recent of 14

Expandability, or the lack thereof, was far and away the most controversial aspect of the original Macintosh hardware design. Apple co-founder Steve Wozniak was a strong believer in hardware expandability, and he endowed the Apple II with luxurious expandability in the form of seven built-in slots for peripheral cards, configured in a clever architecture that allowed each card to incorporate built-in software on its own ROM chip. This flexibility allowed the Apple II to be adapted to a wider range of applications, and quickly spawned a thriving third-party hardware industry.

But Jef Raskin had a very different point of view. He thought that slots were inherently complex, and were one of the obstacles holding back personal computers from reaching a wider audience. He thought that hardware expandability made it more difficult for third party software writers since they couldn't rely on the consistency of the underlying hardware. His Macintosh vision had Apple cranking out millions of identical, easy to use, low cost appliance computers and since hardware expandability would add significant cost and complexity it was therefore avoided.

Apple's other co-founder, Steve Jobs, didn't agree with Jef about many things, but they both felt the same way about hardware expandability: it was a bug instead of a feature. Steve was reportedly against having slots in the Apple II back in the days of yore, and felt even stronger about slots for the Mac. He decreed that the Macintosh would remain perpetually bereft of slots, enclosed in a tightly sealed case, with only the limited expandability of the two serial ports.

Mac hardware designer Burrell Smith and his assistant Brian Howard understood Steve's rationale, but they felt differently about the proper course of action. Burrell had already watched the Macintosh's hopelessly optimistic schedule start to slip indefinitely, and he was unable to predict when the Mac's pioneering software would be finished, if ever. He was afraid that Moore's Law would make his delayed hardware obsolete before it ever came to market. He thought it was prudent to build in as much flexibility as possible, as long as it didn't cost too much.

Burrell decided to add a single, simple slot to his Macintosh design, which made the processor's bus accessible to peripherals, that wouldn't cost very much, especially if it wasn't used. He worked out the details and proposed it at the weekly staff meeting, but Steve immediately nixed his proposal, stating that there was no way that the Mac would even have a single slot.

But Burrell was not that easily thwarted. He realized that the Mac was never going to have something called a slot, but perhaps the same functionality could be called something else. After talking it over with Brian, they decided to start calling it the "diagnostic port" instead of a slot, arguing that it would save money during manufacturing if testing devices could access the processor bus to diagnose manufacturing errors. They didn't mention that the same port would also provide the functionality of a slot.

This was received positively at first, but after a couple weeks, engineering manager Rod Holt caught on to what was happening, probably aided by occasional giggles when the diagnostic port was mentioned. "That things really a slot, right? You're trying to sneak in a slot!", Rod finally accused us at the next engineering meeting. "Well, that's not going to happen!"

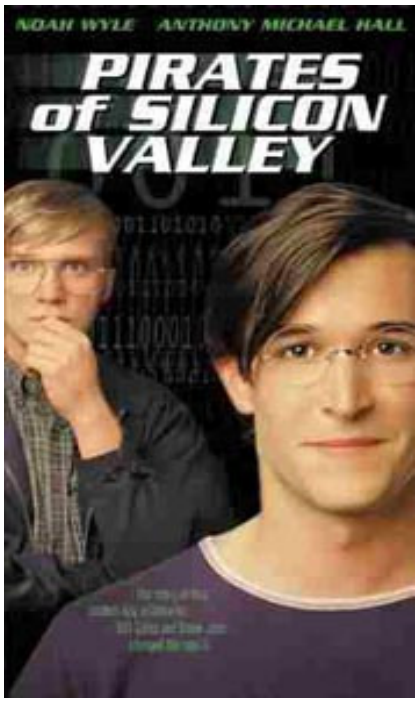
Even though the diagnostic port was scuttled, it wasn't the last attempt at surreptitious hardware expandability. When the Mac digital board was redesigned for the last time in August 1982, the next generation of RAM chips was already on the horizon. The Mac used 16 64Kbit RAM chips, giving it 128K of memory. The next generation chip was 256Kbits, giving us 512K bytes instead, which made a huge difference.

Burrell was afraid the 128Kbyte Mac would seem inadequate soon after launch, and there were no slots for the user to add RAM. He realized that he could support 256Kbit RAM chips simply by routing a few extra lines on the PC board, allowing adventurous people who knew how to wield a soldering gun to replace their RAM chips with the newer generation. The extra lines would only cost pennies to add.

But once again, Steve Jobs objected, because he didn't like the idea of customers mucking with the innards of their computer. He would also rather have them buy a new 512K Mac instead of them buying more RAM from a third-party. But this time Burrell prevailed, because the change was so minimal. He just left it in there and no one bothered to mention it to Steve, much to the eventual benefit of customers, who didn't have to buy a whole new Mac to expand their memory.

# Shut Up!

**Author:** Andy Hertzfeld
**Date:** July 1981
**Characters:** Andy Hertzfeld, Bud Tribble, Steve Jobs, Bill Gates, Charles Simonyi
**Topics:** 3rd party developers, Microsoft, Prototypes
**Summary:** The first time we demoed the Macintosh to Microsoft
**Revision:** most recent of 4




**This movie had a scene based on this story**

Apple had already learned the value of having a thriving third party software market with the Apple II, whose sales increased more than ten-fold when Visicalc, developed by a tiny company called Software Arts, caught on in the business market. The Macintosh intended to replicate the success of the Apple II as an industry standard platform, so it was very important to bring third party developers into the picture as soon as possible.

Microsoft was an obvious choice for one of the first companies for us to talk to. Both companies were started around the same time during the infancy of the personal computer industry, and they already had a business relationship, since Apple licensed Microsoft's Applesoft Basic for the Apple II. Bill Gates and Steve Jobs were born in the same year and shared a similar vision for the potential of personal computers. Each one thought he was smarter than the other one, but Steve generally treated Bill as someone who was slightly inferior, especially in matters of taste and style. Bill looked down on Steve because he couldn't actually program.

Steve had mentioned the Macintosh project to Bill at an industry conference in April, which led to a meeting in Seattle in June, where Steve spun an intriguing vision of pumping out Macintoshes by the millions in an automated factory. The plan was for Microsoft to develop a series of applications for the Macintosh, to be ready at launch. Things went well enough to schedule a meeting in Cupertino in July, where we promised a demo of the actual machine.

Unfortunately, there was one small snag. We were using the Lisa as the development machine for writing the software for Macintosh, and we hadn't yet reached the point where the Macintosh could run stand-alone. The Macintosh needed to be hooked up to a Lisa, in order to download software from it. But the Lisa group was writing all of its own applications for Lisa and didn't want Microsoft to see a Lisa. They made us promise that we wouldn't let Microsoft see the Lisa.

We finally came up with a solution where we'd use a twenty-five foot cable and keep the Lisa in a different room that the Microsoft guys weren't allowed to enter. I would start up the programs on the Lisa in the other room, and Bud Tribble would operate the Macintosh. I would usually run into the main room to see their reaction.

Bill Gates showed up in the early afternoon with three other colleagues: Charles Simonyi, who had recently joined Microsoft from Xerox PARC, Jeff Harbers, who would manage the Macintosh development team, and Mark Matthews, who was to be the technical lead on the project.

They crowded around the prototype, and we started to run our various demos for them, with Steve doing most of the talking.

You could tell that Bill Gates was not a very good listener - he couldn't bear to have anyone explain how something worked to him - he had to leap ahead instead and guess about how he thought it would work.

We showed him how the Macintosh mouse cursor moved smoothly, in a flicker-free fashion.

"What kind of hardware do you use to draw the cursor?", he asked. Many current personal computers had special hardware to draw small bitmaps called "sprites", and he thought we might be doing something similar.

Of course, the Macintosh didn't use any special hardware at all. It did everything in software, which was more flexible anyway, during the vertical blanking interval to eliminate the possibility of flicker. In fact, Burrell and I had recently gotten a mouse to run smoothly on an Apple II, using a similar technique (see Apple II Mouse Card) .

"We don't have any special hardware for it!" I blurted out, probably with a proud sneer in my voice. "In fact..." I was about to mention that we got it running on an Apple II, which had one tenth the processing horsepower of a Macintosh, when Steve guessed what I was about to say.

"Shut up!", he yelled as loud as he could, looking directly at me. He yelled it again, possibly trying to drown me out in case I kept on going. But I understood what he wanted, changing what I was going to say, "In fact, doing it in software is better anyway".

The rest of the demo went pretty well, and both teams shared their excitement about how the Macintosh was going to take the industry to another level. We went out to dinner at a fancy restaurant in Los Gatos to celebrate working together, and we agreed to a framework of a deal where we would give them a stand-alone Mac prototype to develop with in the fall. We were very excited to be working with a third-party company that seemed to understand and appreciate what we were doing.

# Donkey

**Author:** Andy Hertzfeld
**Date:** August 1981
**Characters:** Bill Gates, Neil Konzen, Steve Jobs, Charles Simonyi
**Topics:** Microsoft
**Summary:** We discover who wrote an awkward game for the PC
**Revision:** most recent of 6

**the original IBM PC**

The first version of the IBM PC was introduced in August 1981. Apple responded by running an ad in the Wall Street Journal with the headline "Welcome, IBM. Seriously." Even though he was usually tight with money, Steve Jobs allowed the Mac team to buy an early unit to dissect and evaluate. The day it became available, we ran to the store and purchased one to take back to the lab.

Needless to say, we were not very impressed with it. From the perspective of the Macintosh that we were already in the midst of bringing to life, it seemed like ancient history the day it came out. There was little, if any, Woz-like cleverness in the hardware design, using dozens of extraneous chips without having any cool features. The 8088 was a decent processor compared to the 6502, but it paled next to the 68000 we were using in the Mac.

But the most clunky part of the system was the software. MS-DOS seemed like a clone of an earlier system, CP/M, and even the demo programs lacked flair. It came with some games written in BASIC that were especially embarrassing.

The most embarrassing game was a lo-res graphics driving game called "Donkey". The player was supposed to be driving a car down a slowly scrolling, poorly rendered "road", and could hit the space bar to toggle the jerky motion. Every once in a while, a brown blob would fill the screen, which was supposed to be a donkey manifesting in the middle of the road. If you didn't hit the space bar in time, you would crash into the donkey and lose the game.

We thought the concept of the game was as bad the crude graphics that it used. Since the game was written in BASIC, you could list it out and see how it was written. We were surprised to see that the comments at the top of the game proudly proclaimed the authors: Bill Gates and Neil Konzen. Neil was a bright teenage hacker who I knew from his work on the Apple II (who would later become Microsoft's technical lead on the Mac project) but we were amazed that such a thoroughly bad game could be co-authored by Microsoft's co-founder, and that he would actually want to take credit for it in the comments.

# First day with the Macintosh team

**Author:** Brian Robertson
**Date:** August 1981
**Characters:** George Crow, Burrell Smith
**Summary:** Rod Holt, who remember every technical detail in the world, forgot he hired me
**Revision:** most recent of 2

Being the first "manufacturing" type person on the team, I showed up at Texaco Towers on my first day (August 17, 1981) at 7:30am. Rod showed up about 8:30 and said "who are you". After talking a while he showed me across the hall to my work space, pointed to a bean bag chair, and said "how's that?". Then the desk arrived.

Then George Crow, Burrell Smith and Dan Kotke found out I was there to help and the requests for sourcing started and didn't stop for the next 3 years.

# Joining the Mac Group

**Author:** Bruce Horn

**Date:** September 1981

**Characters:** Alan Kay, Steve Jobs, Andy Hertzfeld, John Warnock, Larry Tesler, Jerry Manock, Rod Holt

**Topics:** Reality Distortion, Apple Spirit, Recruiting

**Summary:** The Reality Distortion Field changes Bruce's mind about working at Apple

**Revision:** most recent of 6

In the spring of 1981, I was 21, and just about to graduate from Stanford. I had been working at least part-time (and often fulltime during the summers) at Xerox PARC in the Learning Research Group for the last eight years. The people at PARC were legendary, and I felt extraordinarily lucky to be able to work with many of them. It was a tremendous learning experience, and I had had the chance to work on a variety of exciting projects through the years.

My most recent project had been the NoteTaker, a portable Smalltalk machine with a bitmap touchscreen display, mouse and keyboard, stereo sound, and dual 8086 processors: one I/O processor which also ran BitBLT (Bitwise BLock Transfer) to draw the graphics, and one emulator processor dedicated to running the Smalltalk bytecode interpreter. My job was to help out on the bytecode interpreter, to write the I/O processor routines, and to basically keep enough of the NoteTaker prototypes running so that they could be used for demos to management. The NoteTaker hardware had been created by Doug Fairbairn, a gifted hardware and chip designer who had recently left to start a new company, VLSI Technology, Inc., or VTI.

On that project I had been working closely with Larry Tesler. Larry was an amazing guy--he had invented the modeless text editing engine for Smalltalk (modeled on his Gypsy editor), and would wear a t-shirt with the slogan "Don't Mode Me In" around the lab. He also was famous for writing a piece of software that would coordinate flash cards at Stanford football games to provide spectators across the stadium with animated bitmap graphics. One evening Larry and I went out for dinner to a local pizza parlor on El Camino in Palo Alto. While we were waiting for the pizza, Larry said, "Bruce, I'm thinking of leaving PARC."

"Really? How can you leave PARC?" I was incredulous. PARC was the Mecca of computer science; we often said (only half-jokingly) that 80 of the 100 best computer scientists in the world were in residence at PARC. I could walk down the hall and wander into the offices of people like Alan Kay (the leader of the Learning Research Group, where many of today's user interface innovations were first created); Chuck Geschke and John Warnock (who later founded Adobe); and Ed McCreight, the inventor of B-Trees. And they all made time to answer my questions, even if they were from a geeky and gangly teenager. LRG and the rest of PARC were full of the brightest and most creative people on the planet. Why would anyone want to leave?

Larry said that he was ready for a new challenge--ready to try to get some of the PARC ideas out into the world. I said, "Well, what about Apple?" We had been talking with Apple recently and had given a demo of Smalltalk to Steve Jobs. Although Larry didn't say anything right away, it turns out he had already been interviewing at Apple, and was soon to join the Lisa team.

When I finally did graduate in mid-1981, I wasn't sure what I wanted to do. Working at PARC was a dream come true, and I had lived that dream for more than a third of my life. But maybe I should get out into the "real world." PARC was so ahead of the rest of the world--we had Alto and Dorado

workstations with mice, large portrait bitmapped displays, WYSIWYG (What You See Is What You Get) text editors, graphics editors, interactive object-oriented programming environments with comprehensive class libraries, local area networks, laser printers, email--that taking a "normal" job in industry was guaranteed to be a letdown. But in talks with Adele Goldberg, one of the senior members of LRG, and my good friend Rachel Rutherford, I realized that I really did need to try something new--that I'd regret it if I didn't leave the comfortable and familiar environment of PARC.

I began interviewing at a variety of companies in the valley; most were forgettable. Then I remembered about Larry Tesler at Apple, and about Doug Fairbairn at VTI, so I contacted them both.

Apple was interesting; Larry was working on the Lisa, which was starting to look like a real computer, but for some reason it didn't appeal to me. At one point, though, Larry realized that I'd be a better fit in the Mac group, and introduced me to Andy Hertzfeld. Andy (the "soul" of the Mac software group) showed me some demos that were so amazing that I somehow thought that they didn't really need me--that the software was almost done! But I was impressed and intrigued, and mulled it over...

Meanwhile, I went to interview at VTI. The people there were wonderful. I'd be working with folks I knew and respected, and Doug even offered me a $15K signing bonus, a huge amount of money for a recent college graduate. I'd be working on advanced chip design tools, a new area for me, and it would be an interesting challenge. So I accepted the job. That was Thursday.

On Friday evening, I got a phone call. "Bruce, it's Steve. What do you think about Apple?" It was Steve Jobs. "Well, Steve, Apple's cool, but I accepted a job at VTI."

"You did what? Forget that, you get down here tomorrow morning, we have a lot more things to show you. Be at Apple at 9am." Steve was adamant. I thought I'd go down, go through the motions, and then tell him that I'd made up my mind and was going to VTI.

Steve switched on the Reality Distortion Field full-force. I met with seemingly everyone on the Mac team, from Andy to Rod Holt to Jerry Manock to the other software engineers, and back to Steve. Two full days of demos, drawings of the various designs, marketing presentations--I was overwhelmed.

On Monday I called Doug Fairbairn at VTI and told him I had changed my mind.

I was going to join Apple, where we would change the world with a little computer called the Macintosh.

# Desk Ornaments

**Author:** Andy Hertzfeld

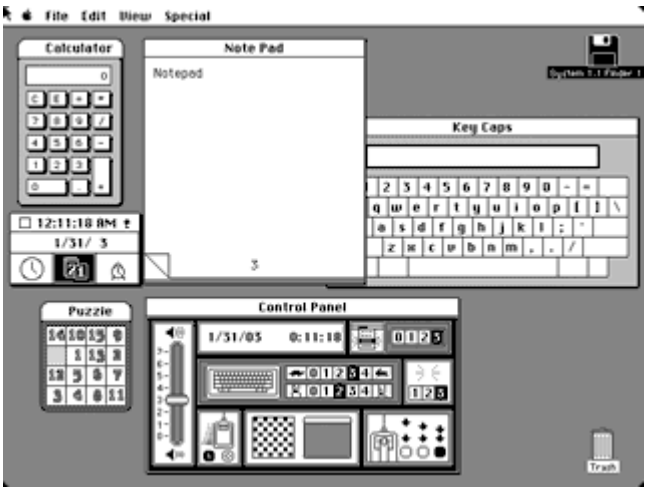
**Date:** October 1981

**Characters:** Bud Tribble, Chris Espinosa, Donn Denman, Larry Kenyon, Steve Capps, Bill Atkinson, Steve Jobs, Susan Kare

**Topics:** Software Design, User Interface, Personality

**Summary:** A brief history of desk accessories

**Revision:** most recent of 21




**Some original Desk Accessories**

One of the first architectural decisions that Bud and I made for the Macintosh system software in the spring of 1981 was that we were only going to try to run one application at a time. We barely had enough RAM or screen space to do even that, and we thought that we'd benefit from the resultant simplifications. Besides, multi-tasking was supposed to be Lisa's forte, and we didn't want to usurp all of the reasons for buying a Lisa.

Bud Tribble was usually on an even keel, but one afternoon in the fall of 1981 he came into my office, unusually excited. "You know, I've been thinking about it. Even if we can only run one major application at a time, there's no reason that we can't also have some little miniature applications running in their own windows at the same time."

That sounded intriguing to me. "What kind of little programs? How are they different?" , I wondered.

Bud smiled. "You'd want tiny apps that were good at a specific, limited function that complements the main application. Like a little calculator, for example, that looked like a real calculator. Or maybe an alarm clock, or a notepad for jotting down text. Since the entire screen is supposed to be a metaphorical desktop, the little programs are desk ornaments, adorning the desktop with useful features."

"But where do we draw the line?", I asked. "What are the differences between an ornament and a real application?"

"Well, ornaments have to fit into the world of the main application", Bud responded, "but not the other way around. The main application owns the menu bar, for example, but maybe the ornaments can have a menu when they're active. The main application would still run its event loop, but it would occasionally pass events to the little guys. And of course you can cut and paste between them."

That sounded good to me, but we had plenty of other stuff to work on first. Bud left Apple in December 1981 to return to medical school, but I explained his concept to new team members and considered them to be part of our plan. Chris Espinosa liked the idea when we explained it to him, and he worked on a visualization of the calculator as part of learning to program with QuickDraw, culminating in a calculator construction set (see Calculator Construction Set) .

Eventually, after getting the initial implementations of the window, menu and event managers going, by May 1982 we had enough of the rudiments done to give desk ornaments a try.

The nascent Mac operating system already had an implementation of loadable device drivers, where installable code could be loaded from disk to manage optional peripherals. I decided to cast desk ornaments as a special kind of driver, adding a few extra calls so they could maintain a window on the screen, receive events from the host application, and occasionally get some processing time to accomplish something. Applications were required to make a system call periodically to yield some time for ornaments to run, and had to occasionally pass events to them and follow a few other conventions for co-existing. The very first desk ornament was a boring one, called "TestOrn", that displayed a rectangular window with an incrementing counter, showing that it was capable of doing background processing, even when it wasn't the top-most window. It wasn't very impressive, but it became the template from which all the others sprung.

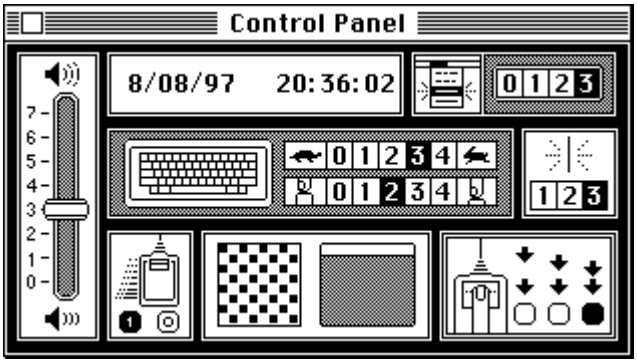
In the fall of 1982, the first pass of the User Interface Toolbox was completed, so I had time to work more on ornaments, which by this time were renamed "desk accessories" by the pubs group who thought that the term "ornament" was too ornamental. I wrote a simple clock, and got the calculator going, modeled after Steve Jobs' design from the calculator construction set. The most interesting one that I wrote in that spurt was a desktop pattern editor, which was a sort of miniature icon editor that let you edit an 8 by 8 pixel pattern by clicking to toggle the dots on and off. When you made a pattern that you liked, you could make it the desktop pattern, filling the background of the screen with your new creation instead of the standard gray.

Within a couple of months, other people wanted to take a crack at writing desk accessories. Someone asked me if they could write one in Pascal, which wasn't easy because you had to pass parameters to the underlying driver in registers, which the Pascal compiler couldn't manipulate directly. I realized that you could write a little bit of standard glue code in assembly language to do the dirty work, and have most of your desk accessory written in Pascal. I wrote a number puzzle in Pascal as a test case, to prove that it could be done (see The Puzzle) .

But once again I had to get back to other things and the desk accessories languished. But by the spring of 1983, the ROM was beginning to stabilize, and we realized that we had to get serious about the desk accessories that would ship with the initial system. Jerome assigned Donn Denman, who was taking a hiatus from working on Basic to help us finish the system, to help with desk accessories. Donn and I collaborated on finishing the calculator, with Donn doing the back-end that interfaced with the floating point routines. Donn was also responsible for the alarm clock and notepad.

Now that we had cut and paste working, I realized that it would be useful to have a desk accessory that could hold commonly used snippets of text and graphics, so they could be pasted into any application. I wrote a desk accessory called the "Gallery" that allowed you to manage a set of clippings using cut and paste, building it on top of the resource manager so it was easy to implement. Everyone seemed to like the functionality but not the name. I was explaining the situation to Larry Kenyon when he suggested that we call it the "Scrapbook", which was perfect, so the Scrapbook joined our growing armada of useful desk accessories.

The Macintosh's character encoding scheme allocated the upper half of the 8-bit character set to occasionally useful but obscure characters that were not portrayed on the keys, accessed with the option key, that were nearly impossible to remember. Steve Capps conceived of a desk accessory called "Key Caps" (named partially after himself?) that displayed a picture of the keyboard, with the keycaps changing depending on the meta-keys that were pressed, which allowed you to hunt for the special characters visually. He coded it up quickly and it became another great addition.




**The Control Panel**

We needed a way to control various system parameters like the sound volume, or the mouse scaling parameters. We decided that a desk accessory would be perfect for that, since it would be easy to access no matter what application you were in. So the last desk accessory that I worked on before shipping, in November 1983, was the Control Panel. Susan Kare came up with a beautiful, highly graphical design (with no text whatsoever) that I implemented using a separate purgable resource for each section, so they didn't have to be in memory at once. It had a little rabbit and tortoise to represent a range of speeds, and lots of other graphical embellishments; after the Mac was released, one review described it as a crib toy, which I took more or less as a compliment.

The most controversial part of the Control Panel was the desktop pattern editor, which I had rescued from its earlier stand-alone incarnation. Users could select from a couple dozen prefabricated desktop patterns or edit their own in fat bits. It was this latter capability that caused problems, as it was pretty easy to create ugly, abominable patterns.

Bill Atkinson complained to me that it was a mistake to allow users to specify their own desktop patterns, because it was harder to make a nice one than it looked, and led directly to ugly desktops. But I thought that users should be free to do as they pleased, since it was their desktop, and it was easy to revert to one of the built-in patterns. Bill cared most about MacPaint, and didn't want a potentially ugly desktop pattern marring his creation. So he made MacPaint allocate a window that was the size of the screen when it started up, and filled it with the standard 50% gray pattern, making his own desktop covering up the real one, thus protecting the poor users from their rash esthetic blunders, at least within the friendly confines of MacPaint.

# I Don't Have a Computer!

**Author:** Bruce Horn
**Date:** December 1981
**Characters:** Andy Hertzfeld, Bruce Horn, Steve Jobs, John Couch
**Topics:** Apple Spirit, Lisa, Lisa Rivalry, Buildings, Management
**Summary:** Bruce needs to find a computer, and fast.
**Revision:** most recent of 2

When I joined the Mac group, we were still very small, working in what we called Texaco Towers--a nondescript office building behind a Texaco station at the corner of De Anza and Stevens Creek, in Cupertino, long since gone (see Texaco Towers) .

The software group had offices on one side of the office building, and the hardware group was on the other side. Our offices were quite spare. One room toward the front of the building had a line printer (one of those old, big, noisy impact printers) that was driven by an Apple II. To print out your sources, you had to write them to an Apple II disk, run a program to swap the bytes or do some other manipulation, and then put it in the Apple II which would then run the line printer.

Of course, to print out sources, you had to write some code. I was ready to go, and I had been on the Mac team for several weeks but still hadn't gotten a Lisa, the development machine we used. The Lisa was still under development, and the prototypes were very difficult to obtain. I was starting to get a little frustrated and complained to Andy Hertzfeld at one point that maybe I should go work for a *real* company that could provide the appropriate tools for their software developers.

Andy mentioned this to Steve Jobs. Steve immediately said, "Well, should we fire him?" Andy replied "No, just get him a computer!"

Later that day I received a note from Steve to go to a particular office in another Apple building; there would be a Lisa on the desk, and I could take that one and use it. I went over to the other building and found the office.

The nameplate on the office door said, "John Couch." John was the head of the group developing the Lisa. I wasn't sure that this was really OK, but I had a note from Steve if anybody asked, so I walked into the office, unplugged the computer, and carted it away.

I still don't know to this day whether Steve had arranged this with John, or if John came back to the surprise of an empty desk, but I did get a lot of use out of that machine, maybe more than John ever did.

# The Grand Unified Model (1) - Resources

**Author:** Bruce Horn

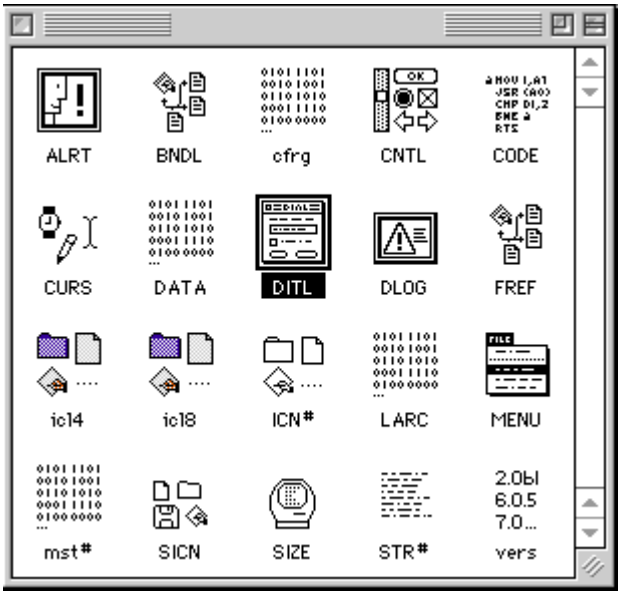
**Date:** December 1981

**Characters:** Andy Hertzfeld, Bob Belleville, Bruce Horn, Alan Kay, Steve Capps, Bill Atkinson, Dan Smith, Frank Ludolph, Larry Kenyon, Joanna Hoffman, Ted Kaehler, Steve Jobs

**Topics:** Software Design, Origins, Technical, User Interface

**Summary:** The creation of resources

**Revision:** most recent of 22




**Applications contained many resources**

Imagine the challenge: designing and implementing a brand new, graphical user interface, operating system, and core applications for a small personal computer to compete with the IBM PC. That's what we were going to do with the Macintosh.

The year was 1981, and I had just graduated from Stanford University. The past eight years I had spent as much time as possible working in the Learning Research Group at the Xerox Palo Alto Research Center. PARC was the place where everybody who was anybody in computer science wanted to go work, or at least visit. And I had just decided to leave PARC to join Apple (see Joining the Mac Group) .

I was used to working on Altos and Dorados, powerful workstations with bitmapped displays, mice, networking, and the most advanced software development system of the time, Smalltalk. These machines had lots of memory, at least 3MB of fast disk storage (remember, this was 1981), and were very fast. To build one of the Altos cost over $15,000, and Dorados much more. At Apple, the Macintosh was going to be a much more limited machine, with a tiny bitmapped display, no hard disk, and a mere smidgen of memory. It would have to be able to be built for under $1000 if we were to compete with the IBM PC and other personal computers of the time. And we were going to try to build a graphical user interface and operating system that would work acceptably on an 8Mhz 68000 microprocessor.

Fortunately, I had worked on a 8086-based system at PARC, the NoteTaker (see Joining the Mac Group) , so I knew what the limitations of small systems were. Still, I was used to the Smalltalk user experience, and wanted to do whatever I could with the Finder to approach the friendliness, flexibility, and ease of use that Smalltalk provided.

Smalltalk was an integrated object-oriented language and programming environment, decades ahead of its time. The class library (one of the first of its kind) was extremely sophisticated, with high-level frameworks for interface support, windowing, and graphics. Programming in Smalltalk was (and still is, in many ways) unlike any other system; making changes to code was fast and easy, and the turnaround time was very short, since the system incrementally incorporated changes as you made them. I was spoiled by the availability of well thought-out class library code that could be leveraged, and by the flexibility and ease with which new code could be written and existing code changed.

But now I was facing a new reality. Macintosh programs would be written in assembly code and Pascal, in a traditional edit-compile-link-debug cycle, and the dynamic nature of a system like Smalltalk was nowhere to be found. And my project was to be the Finder, the program that every Macintosh user would interact with to organize and manage their files, folders, and applications, which would be the "face" of the Macintosh.

The rigid nature of traditional programming bothered me. The idea of an application frozen in code, with no way to change anything dynamically, was anathema to my ideals. I wanted to be able to change as much as possible at runtime. Of course, the application code itself couldn't be changed, but what could be changed without having to recompile the code? The Finder, in particular, was going to need to maintain dynamic information about the files on the disk, and their relationships to the programs that created them and some of this information, such as a program's application and document icons, would have to be extracted from the program itself. How would this information be stored?

Alan Kay always said that any problem in Computer Science could be solved by adding another level of indirection. I thought that if we could refer to the program *data* separately and indirectly -- the strings, bitmaps, window and dialog layouts, and other non-code information -- we could make it possible for this information to be changed by people who would not have access to the source code. These people -- translators, artists, and designers -- would be able to change the text strings (to translate menu items from English to Norwegian, for example), modify the application and document icons, and replace graphical elements in the program, if the program were written such that these items were factored out of the application.

The ability to easily localize applications and the operating system would be novel, especially in the early 1980's. None of the systems that I had used, including Smalltalk, had this ability; it was just assumed that everybody using the system would be English-speaking, and that other countries would be building their own systems. If the Mac were able to be released in other countries, with menus, icons, dialogs, dates, and sorting orders translated to different languages, it would make a big improvement in our potential market share. I can't even remember when I started to recognize that the localization ability was necessary; it was a meme (probably started by Joanna Hoffman) that infected us all in the Mac group.

I figured that the Mac should provide this facility for all developers as a part of the User Interface Toolbox: a way to both provide for factoring out non-program data to make it easy to write modifiable applications, as well as manage memory on a fine-grained basis, object by object. Thus the idea for Resources was born.

**Resources**

The Resource Manager was a solution to several problems: managing dynamic data for the Finder; factoring out localizable information (strings, icons, and so on) from applications, and finally, managing memory use as frugally as possible. The Mac had relatively little memory, only 128K, for both the system and whatever application was running. The floppy drive was only 400K, and would have to store the system, applications, and documents, so anything that could reduce the amount of space required, by sharing data, would have a big impact. Graphics were notoriously memory-hungry, and the Finder, in particular, would have to juggle icons, swapping their bitmaps in and out of memory as required. The Resource Manager would have to handle each of these small entities separately. Thankfully, I knew of a similar system in Smalltalk, an object-oriented virtual memory called OOZE that was designed by Ted Kaehler, that swapped objects in and out of main memory as required. This was my inspiration for the Resource Manager. Find out more about OOZE here .

Like OOZE, the Resource Manager maintained objects in a resident object table, called the resource map. A resource file was the on-disk representation of the resources. Resources would be typed: a menu object would be of type MENU, and a text string would be of type TEXT. Resource types were four bytes, to fit within a single 32-bit word as well as to be somewhat human-readable. All resources would have a resource ID, unique within the type domain of the resource file, and any resource could also be named, for the convenience of the programmer.

As it happened, our memory manager provided a mechanism that the Resource Manager could leverage for in-memory storage of a resource object: the handle. A handle was a relocatable block of memory, referenced by an indirect pointer (a pointer to a pointer). To allow resources to be purged from memory, blocks could be reclaimed, releasing memory for other uses, while maintaining the original indirect pointer which would then point to a nil value, indicating that there is no block of memory reserved. This allowed objects to be referenced by unique handles whether or not they were currently in memory, greatly simplifying the task of managing dynamically-loadable objects.

I wanted to make sure that we could leverage resources throughout the system, but with maximum flexibility. I felt that it was important that there be a mechanism for system resources to be overridden by the application, and for application resources, in turn, to be overridden by the application's open documents. We could then have documents that carried their own fonts, images, and so on; applications with application-specific resources, such as (again) fonts and images; and system resources that would be shared by all applications. So, resource files would be linkable, staging resource lookup: searching first the document, then the application, and finally, the system.

One of my faults as a software designer is that I seem to try to solve the biggest problems, and to do it bottom-up; that is, I end up doing lots and lots of work with no apparent results, as I build the foundation for the rest of the system. I ended up making the Finder dependent on the Resource Manager, which, while logical in fact, unnecessarily made things a bit more exciting than they otherwise would have been in the development of the Mac, as the Finder's development was pushed out later and later in the schedule. And of course, I didn't come up with these ideas all at once, but as I gained experience with the idea of resources, and as others in the group began to use resources in the rest of the system, the features needed became apparent. And the concepts behind the Resource Manager widely influenced the handling of data throughout the Mac in a sort of Grand Unified Model.

When I started talking with the rest of the team, Larry Kenyon and Andy Hertzfeld realized immediately the importance of resources, and realized that they'd have to make changes in the rest of the system to take advantage of the Resource Manager. Larry was writing the memory manager, and Andy was writing much of the rest of the Toolbox; both were rather far along and the Toolbox, in particular, needed significant changes to use resources. We were all in agreement...but I failed to mention this to Bob Belleville, thinking that it wouldn't take that long to finish anyway, so why bother?

I started writing the code. Some weeks later, Bob Belleville asked me what I was doing, and I told him about the Resource Manager and why we needed it. He told me to stop working on it. I was flabbergasted. I tried to explain to him that the Resource Manager was the keystone of the entire data architecture of the Mac--that the Toolbox, Segment Loader, and the Finder were all dependent on it, and I couldn't think of a way to solve all the problems we needed to solve without something like the Resource Manager.

Bob told me to abandon the project, that we didn't need it, and that I should focus on the Finder (see Resource Manager Countdown) . Of course, the way I planned to fit all the pieces together with the Finder also required the Resource Manager, and I thought it would take much longer to try to code the system without it. I told Bob that I was going to do it anyway. (see You Can't Fire Bruce!) .

It was quite a job, since we only had 64K bytes of ROM for the entire Toolbox. By writing in assembly code, I was able to implement the complete Resource Manager in only 3K bytes of code, leaving a few bytes left over in the ROM for the inevitable bug fixes and feature additions. But it had taken quite a bit longer than I had hoped it would, and the delay put time pressure on all of us.

We were lucky: it turned out that resources were the answer to quite a few questions we hadn't known to ask. Andy used the Resource Manager to swap code segments dynamically, so that large programs could run in the tiny amount of RAM; he rewrote the Font, Menu, and Control managers to store and retrieve their data from resource files; and more and more shared data in the system ended up being stored as resource objects. Because it was relatively simple to use resources to store text strings and menus in the system and in applications, it became much easier to provide localized systems for different languages and countries. Joanna Hoffman and Alain Rossmann designed a resource-based system for handling special time and date formats for different countries, and the Mac became a truly multilingual system.

Almost every piece of data in the Macintosh ended up being touched by the Grand Unified Model. Even transient data, data being cut and pasted within and between applications, did not escape. The Scrap Manager labeled each piece of data on the clipboard with a resource type. In another Mac innovation, multiple pieces of data, each of a different type, could be stored on the clipboard simultaneously, so that applications could have a choice of representation of the same data (for example, storing both plain and styled text). And since this data could easily be stored on disk in a resource file, we were able to provide cutting and pasting of relatively large chunks of data by writing a temporary file called the Clipboard.

So, in late 1982, with the Resource Manager finished, more or less, I finally turned my attention back to the Finder (see The Grand Unified Model (2) - The Finder) .

# Hungarian

**Author:** Andy Hertzfeld
**Date:** January 1982
**Characters:** Bud Tribble, Tom Malloy, Charles Simonyi, Bill Atkinson
**Topics:** Software Design, Technical, Lisa
**Summary:** A curious style of programming
**Revision:** most recent of 13

**Charles Simonyi in the early days**

The Macintosh used the same Motorola 68000 microprocessor as its predecessor, the Lisa, and we wanted to leverage as much code written for Lisa as we could. But most of the Lisa code was written in the Pascal programming language. Since the Macintosh had much tighter memory constraints, we needed to write most of our system-oriented code in the most efficient way possible, using the native language of the processor, 68000 assembly language. Even so, we could still use Lisa code by hand translating the Pascal into assembly language.

We directly incorporated Quickdraw, Bill Atkinson's amazing bit-mapped graphics package, since it was already written mostly in assembly language. We also used the Lisa window and menu managers, which we recoded in assembly language from Bill's original Pascal, reducing the code size by a factor of two or so. Bill's lovely Pascal code was a model of clarity, so that was relatively easy to accomplish.

The Mac lacked the memory mapping hardware prevalent in larger systems, so we needed a way to relocate memory in software to minimize fragmentation as blocks got allocated and freed. The Lisa word processor team had developed a memory manager with relocatable blocks, accessing memory blocks indirectly through "handles", so the blocks could be moved as necessary to reduce fragmentation. We decided to use it for the Macintosh, again by recoding it from Pascal to assembly language.

The primary author of the Lisa word processor and its memory manager was Tom Malloy, an original member of the Lisa team and Apple's first recruit from Xerox PARC. Tom had worked on the Bravo word processor at PARC under the leadership of Charles Simonyi, and used many of the techniques that he learned there in his Lisa code.

Even though Bud Tribble had to leave the Mac team in December 1981 in order to retain his standing in the M.D./Ph.D. program at the University of Washington, he decided that he could still do the initial implementation of the memory manager, as we were planning all along, hoping to finish it quickly after he moved back to Seattle, before classes started. He obtained a copy of the memory manager source from Tom Malloy, but he was in for a shock when he began to read the code.
The memory manager source lacked comments, which was disappointing, but the biggest obstacle was the names selected for variables and procedures: all the vowels were gone! Every identifier seemed to be an unpronounceable jumble of consonants, making it much harder to understand the code, since a variable's meaning was far from obvious. We wondered why the code was written in such an odd fashion. What happened to all of the vowels?

It turns out that Tom Malloy was greatly influenced by his mentor at Xerox, a strong-willed, eccentric programmer named Charles Simonyi. Charles was quite a character, holding many strong opinions about the best way to create software, developing and advocating a number of distinctive coding techniques, which Tom brought to the Lisa team. One of the most controversial techniques was a particular method of naming the identifiers used by a program, mandating that the beginning of each variable name be determined by the type of the variable.

However, most of the compilers in the early eighties restricted the length of variable names, usually to only 8 characters. Since the beginning of each name had to include the type, there weren't enough characters left over to use a meaningful name describing the purpose of the variable. But Charles had a sort of work-around, which was to leave out all of the vowels out of the name.

The lack of vowels made programs look like they were written in some inscrutable foreign language. Since Charles Simonyi was born and raised in Hungary (defecting to the west at age 17), his coding style came to be known as "Hungarian". Tom Malloy's memory manager was an outstanding specimen of Hungarian Pascal code, with the identifiers looking like they were chosen by Superman's enemy from the 5th dimension, Mr. Mxyzptlk.

Bud decided that it would be too error prone to try to translate the Hungarian memory manager directly into assembly language. First, he made a pass through it to strip the type prefixes and restore the vowels to all the identifier names, so you could read the code without getting a headache, before adding lots of block comments to explain the purpose of various sub-components.

A few weeks later, when Bud came back to attend one of our first retreats, he brought with him a nicely coded, efficient assembly language version of the memory manager, complete with easy to read variable names, which immediately became a cornerstone of our rapidly evolving Macintosh operating system.

# -2000 Lines Of Code

**Author:** Andy Hertzfeld
**Date:** February 1982
**Characters:** Bill Atkinson
**Topics:** Software Design, Management, Lisa
**Summary:** It's hard to measure progress by lines of code
**Revision:** most recent of 1

In early 1982, the Lisa software team was trying to buckle down for the big push to ship the software within the next six months. Some of the managers decided that it would be a good idea to track the progress of each individual engineer in terms of the amount of code that they wrote from week to week. They devised a form that each engineer was required to submit every Friday, which included a field for the number of lines of code that were written that week.

Bill Atkinson, the author of Quickdraw and the main user interface designer, who was by far the most important Lisa implementor, thought that lines of code was a silly measure of software productivity. He thought his goal was to write as small and fast a program as possible, and that the lines of code metric only encouraged writing sloppy, bloated, broken code.

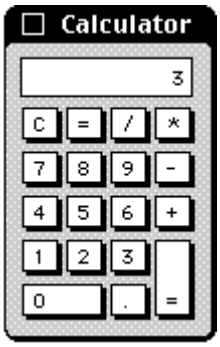
He recently was working on optimizing Quickdraw's region calculation machinery, and had completely rewritten the region engine using a simpler, more general algorithm which, after some tweaking, made region operations almost six times faster. As a by-product, the rewrite also saved around 2,000 lines of code.

He was just putting the finishing touches on the optimization when it was time to fill out the management form for the first time. When he got to the lines of code part, he thought about it for a second, and then wrote in the number: -2000.

I'm not sure how the managers reacted to that, but I do know that after a couple more weeks, they stopped asking Bill to fill out the form, and he gladly complied.

# Calculator Construction Set

**Author:** Andy Hertzfeld
**Date:** February 1982
**Characters:** Chris Espinosa, Steve Jobs, Donn Denman
**Summary:** Chris tries to make a Steve-approved calculator
**Revision:** most recent of 2



**The Calculator**

Chris Espinosa was one of Apple's earliest and youngest employees, who started work for the company at the ripe age of 14. He left Apple in 1978 to go to college at UC Berkeley, but he continued to do freelance work during the school year, like writing the Apple II Reference Manual, the replacement for the legendary "Red Book".

In the summer of 1981, Steve Jobs convinced Chris to drop out of school to come work on the Mac team full time, arguing that he could go back to school anytime, but there'd only be one chance to help shape the Macintosh. Chris dropped out of school to become the manager of documentation for the Macintosh, starting in August 1981.

We needed technical documentation right away, since we planned to seed third party developer in only a few months. Since the most important part of the Macintosh software was the Quickdraw graphics package, Chris decided to start with documenting Quickdraw.

Chris wanted to write a demo program using Quickdraw, in order to better understand it. He got excited about the idea of what we called "desk ornaments", which at that point were not implemented yet. He decided to work on a Quickdraw program to draw the calculator.

After playing around for a while, he came up with a calculator that he thought looked pretty good. But the acid test was showing it to Steve Jobs, in his role as our esthetic compass, to see what he thought.

We all gathered around as Chris showed the calculator to Steve and then held his breath, waiting for Steve's reaction. "Well, it's a start", Steve said, "but basically, it stinks. The background color is too dark, some lines are the wrong thickness, and the buttons are too big." Chris told Steve he'll keep changing it, until Steve thought he got it right.

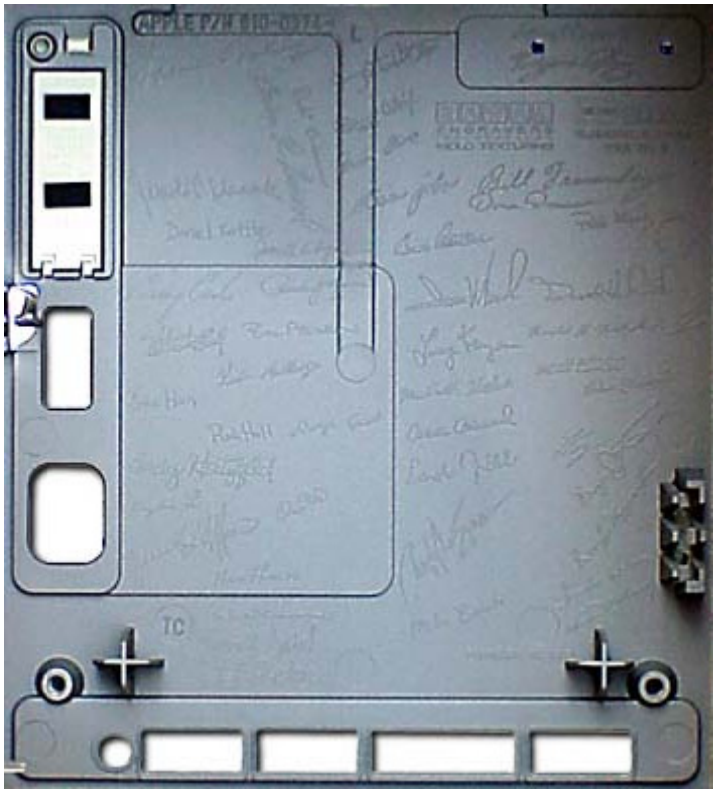
So, for a couple of days, Chris would incorporate Steve's suggestions from the previous day, but Steve would continue to find new faults each time he was shown it. Finally, Chris got a flash of inspiration.

The next afternoon, instead of a new iteration of the calculator, Chris unveiled his new approach, which he called "the Steve Jobs Roll Your Own Calculator Construction Set". Every decision regarding graphical attributes of the calculator were parameterized by pull-down menus. You could select line thicknesses, button sizes, background patterns, etc.

Steve took a look at the new program, and immediately started fiddling with the parameters. After trying out alternatives for ten minutes or so, he settled on something that he liked. When I implemented the calculator UI (Donn Denman did the math semantics) for real a few months later, I used Steve's design, and it remained the standard calculator on the Macintosh for many years, all the way up through OS 9.

# Signing Party

**Author:** Andy Hertzfeld
**Date:** February 1982
**Characters:** Steve Jobs, Bob Belleville, Jerry Manock, Mac Team, Bud Tribble, Jef Raskin, Mike Boich, Rod Holt, Steve Wozniak, Mike Murray
**Topics:** Management, Apple Spirit, Industrial Design
**Summary:** The artists sign their work
**Revision:** most recent of 6



**the back side of the Macintosh case**

The component of the Macintosh hardware that had the longest lead time was the hard tool that molded its distinctive plastic case. After tweaking the case design for more than six months and building a small production run of 50 units with a soft-tooled case, the final design was ready to go out for hard tooling toward the end of February 1982, so we could meet the ship date that we were aiming for at the time, which was January 1983.

The Mac team had a complicated set of motivations, but the most unique ingredient was a strong dose of artistic values. First and foremost, Steve Jobs thought of himself as an artist, and he encouraged the design team to think of ourselves that way, too. The goal was never to beat the competition, or to make a lot of money; it was to do the greatest thing possible, or even a little greater. Steve often reinforced the artistic theme; for example, he took the entire team on a field trip in the spring of 1982 to the Louis Comfort Tiffany museum, because Tiffany was an artist who learned how to mass produce his work.

Since the Macintosh team were artists, it was only appropriate that we sign our work. Steve came up with the awesome idea of having each team member's signature engraved on the hard tool that molded the plastic case, so our signatures would appear inside the case of every Mac that rolled off the production line. Most customers would never see them, since you needed a special tool to look inside, but we would take pride in knowing that our names were in there, even if no one else knew.

**The signature sheet**

We held a special signing party after one of our weekly meetings on February 10, 1982. Jerry Mannock, the manager of the industrial design team, spread out a large piece of drafting paper on the table to capture our signatures. Steve gave a little speech about artists signing their work, and then cake and champagne were served as he called each team member to step forward and sign their name for posterity. Burrell had the symbolic honor of going first, followed by members of the software team. It took forty minutes or so for around thirty-five team members to sign. Steve waited until last, when he picked a spot near the upper center and signed his name with a flourish.

We were aware that the team was still growing rapidly, and in a few months there would be a new crop of key contributors that also deserved to sign the case. We decided to draw the line at the date of the signing party, and not to let new signatures come in later, but we knew it would be tough to stick to that. We also wanted to add the signatures of a few major contributors who had left the project: Steve Wozniak, Jef Raskin and Bud Tribble. But that was supposed to be it.

Over the next few months, a few more signatures of people who weren't on the team at the time of the signing party managed to make it into the case. For a while Rod Holt held the line, but eventually Bob Belleville, who hired on in April 1982 as the software manager but soon became the overall engineering manager when Rod Holt retired, decided to add his own name. He also snuck in a few other key people, like marketing manager Mike Murray and original evangelist Mike Boich, who started before he did and who otherwise would have just missed the cut-off.

And then, over time, names gradually began to disappear for practical reasons, as Apple changed the case to make it easier to manufacture. Some details were changed even before first ship, partially obscuring some of the signatures. Each time the case was revised, more names were left off, as dictated by the nature of the revision, until a substantial number of them were gone. I'm not sure which model was the last to have any names at all, but I'm pretty sure that the Macintosh Classic, from the early nineties, didn't have any left.

# Mister Macintosh

**Author:** Andy Hertzfeld
**Date:** February 1982
**Characters:** Steve Jobs, Andy Hertzfeld, Susan Kare, Folon
**Topics:** Software Design, Personality
**Summary:** Steve has a unique idea for the software
**Revision:** most recent of 3

**Mr. Macintosh,**
**as rendered by Folon (courtesy of Digibarn)**

Steve Jobs often came by Texaco Towers after dinner, to see what was new, and we'd usually show him whatever recent progress we made. Sometimes he'd be pissed off about something, but other times he'd be really excited about a new idea.

I was the only one in the office one evening when he burst in, exclaiming that he had a flash of inspiration.

"Mr. Macintosh! We've got to have Mr. Macintosh!"

"Who is Mr. Macintosh?", I wondered.

"Mr. Macintosh is a mysterious little man who lives inside each Macintosh. He pops up every once in a while, when you least expect it, and then winks at you and disappears again. It will be so quick that you won't be sure if you saw him or not. We'll plant references in the manuals to the legend of Mr. Macintosh, and no one will know if he's real or not."

Engineers like myself always daydream about building surreptitious little hacks into the software, but here was the co-founder and chairman of the company suggesting something really wild. I enthusiastically pressed him for details. Where should Mr. Macintosh appear? How often? What should he do when he shows up?

"One out of every thousand or two times that you pull down a menu, instead of the normal commands, you'll get Mr. Macintosh, leaning against the wall of the menu. He'll wave at you, then quickly disappear. You'll try to get him to come back, but you won't be able to."

I loved the idea and promised that I would implement Mr. Macintosh, but not right away, since there were still so many more basic things to get done. Steve told the idea to the marketing team, and eventually recruited the French artist Folon to do some renditions of Mr. Macintosh. I also asked my high school friend Susan Kare, who hadn't started with Apple yet, to try to draw some Mr. Macintosh animations.

Most of the Macintosh system software had to be packed into a 64 KByte ROM, and ROM space got more scarce as development proceeded and the system grew. Eventually, it was clear that we'd never be able to fit bitmaps for Mr. Macintosh into the ROM, but I wasn't willing to give up on him yet.

I made the software that displayed the menus look at a special low memory location called the "MrMacHook", for an address of a routine. If the routine is present, it's called with parameters that let it draw in the menu box, and it returns a result that tells the menu manager if it did anything. Using this, an application or system module could implement Mr. Macintosh (or perhaps his evil twin) if they saw fit.

I'm not sure if anybody ever actually implemented Mr. Macintosh or used the "MrMacHook" for something worthwhile.

# Rosing's Rascals

**Author:** Andy Hertzfeld
**Date:** March 1982
**Characters:** Bill Atkinson, Wayne Rosing, Dan Smith, Frank Ludolph, Steve Jobs, Bruce Horn
**Summary:** The Lisa Filer was radically redesigned with no time to spare

By the spring of 1982, the Lisa User Interface was finally settling down, and the software team was working feverishly to get everything ready to ship by their deadline in the fall. Most of the applications were shaping up, although myriad problems remained, and the team could finally sense a glimmer of light at the end of the long tunnel.

Dan Smith and Frank Ludolph were working on the Lisa Filer, the key application that managed files and launched other applications. It was beginning to come together, but Dan was still unsatisfied with the current design.

The Filer was based on a dialog window that prompted the user to select a document from a list, and then select an action like "Open", "Copy" or "Discard", and then answer more questions, depending on the selected action. There was so much prompting that it became known as the "Twenty Questions Filer". Dan thought that it wasn't easy or enjoyable to use, but there just wasn't enough time left in the schedule for further experimentation, so they were pretty much stuck with it.

One afternoon, Dan mentioned his dissatisfaction to Bill Atkinson, the main designer of the Lisa User Interface. Bill suggested that they meet that evening at his home in Los Gatos for a brain-storming session to see if they could come up with a better design, even though it was probably too late to use it for the initial release.

Bill favored a more graphical approach, and wanted to use small graphical images to represent files, which could be manipulated by dragging them with a mouse. He remembered an interesting prototype that he saw at M.I.T. called Dataland, where data objects could be spatially positioned over a large area. He adapted the idea for Lisa, allowing icons representing files and directories to be positioned on a scrolling, semi-infinite plane.

After a couple of nights of fiddling around, Dan and Bill had an interesting mock-up going, with icons representing documents and folders, complete with a trash can with flies buzzing around it. The icons used a mask bitmap to define their borders, so irregular shapes could be rendered seamlessly on the gray desktop. The new design seemed to have the simplicity and elegance that they were striving for, so they began to get excited.

They were both eager and afraid to show the mock-up to the rest of the team, since the design of the Filer was supposed to be frozen, and embarking on such a major revision would surely slip the schedule, which was already precariously close to unrealizable. They gathered up their courage and approached Wayne Rosing, the Lisa Engineering Manager, and explained their dilemma.

Wayne appreciated the potential of the new approach, but wasn't ready to slip the schedule to accommodate it. He thought it was perhaps barely possible to go with both the new design and the current schedule, if they could turn the mock-up into a solid working prototype in record time. He proposed a deal: he gave them permission to work on the new design in secret for the next two weeks. If they had a robust, stable prototype by then, he promised to support it. If they didn't, Bill and Dan promised to forget it and work to finish the earlier design.

Wayne extracted one additional promise from Bill: under no circumstances was he to show the mock-up to Steve Jobs. Wayne knew that Steve would have a strong reaction and would probably wreak havoc with the schedule accordingly. He didn't want Steve to see it until they knew if they could pull it off.

Bill was used to showing off his latest advances to the Mac team, and this new, icon-based approach to file management was a particularly important one. Bruce Horn had started working on the Mac team the previous month, and he was already starting to develop our file manager, which Bud had christened "The Finder". Bruce had similar ideas about spacial filing, and he and I had created a prototype called the "micro-finder" which represented files as tabs that were spacially organized on a picture of a diskette. Bill thought it was important for us to see the new direction as soon as possible, so he left us a copy of his prototype, under strict instructions not to show it to Steve.

We had a few close calls over the next couple of weeks as we played with the prototype, frantically quitting it when we heard Steve approaching. Finally, on the last day before the deadline expired, we must have cut it too close, because Steve knew that we were hiding something from him. We explained our promise to Bill, but Steve still demanded to see it, so we had to show it to him. He immediately fell in love with it, and ran off to talk to Bill and Wayne about it, just as we feared.

Luckily, the development had gone well the last two weeks, and Wayne was ready to commit to the new approach and unveil it to the entire team. He called an all-hands meeting, to which Bill, Dan, Frank and Wayne wore newly minted T-Shirts labeled "Rosing's Rascals". Wayne explained the surreptitious nature of the two week effort to the team while Bill set up the demo. Rosing's rascals had pulled it off, endowing the Lisa with a much more intuitive file manager that quickly became a hallmark of Apple's new user interface.

# Software Wizard

**Author:** Andy Hertzfeld
**Date:** March 1982
**Characters:** Andy Hertzfeld, Burrell Smith, Steve Jobs
**Topics:** Personality, Apple Spirit
**Summary:** The Mac group gets business cards
**Revision:** most recent of 1

By the spring of 1982, the Macintosh project was considered more legitimate within Apple. It was beginning to transition from a research effort into a mainstream project. We had to get more organized as the team grew.

Initially, we didn't have formal titles in the Mac group, but we needed to figure out what they were in order to get business cards made. My title with the Apple II group was "Senior Member of Technical Staff", which sounded dull to me. I told Peggy Alexio, Rod Holt's secretary, who was ordering the business cards, that I didn't want any, because I didn't like my title.

The next day Steve came by and told me that he heard that I didn't want business cards, but he wanted me to have them, and he didn't care what title I used; I could pick any title that I liked. After a little bit of thought, I decided on "Software Wizard", because you couldn't tell where that fit in the corporate hierarchy, and it seemed a suitable metaphor to reflect the practical magic of software innovation.

When I told Burrell about my new title, he immediately claimed "Hardware Wizard" for himself, even though I discouraged him, since it diminished the uniqueness of my title. And, as soon as word got around, lots of other folks on the Mac team started to change their titles to something more creative. The trend persisted at Apple for many years, and even spread to other companies, but as far as I know, that's how it got started.

# Gobble, Gobble, Gobble

**Author:** Andy Hertzfeld
**Date:** March 1982
**Characters:** Steve Jobs, Andy Hertzfeld, Burrell Smith, Angeline Lo
**Summary:** We interview candidates for software manager
**Revision:** most recent of 3

When Bud told us in early December of 1981 that he had to leave the Mac team to go back to Seattle to keep his place in medical school, both Burrell and I were pretty shaken. We were worried that we couldn't pull it off without Bud and that we'd get some authoritarian manager instead, who would wreck the unique spirit of our team. We expressed our concerns to Steve, who promised us that we'd have a big say in hiring the new manager, and that he would personally protect us if a situation like the one we were fearing ever arose.

In January, we began interviewing candidates for the software manager position. We had high standards and expectations and interviewed a number of outstanding people, like Ed Taft (who became one of the first employees at Adobe) and Tim Mott (who helped start Electronic Arts), who for one reason or another didn't take the job. But we also interviewed some more mundane candidates, which could sometimes get pretty wild if Steve decided he didn't respect someone.

For example, Burrell, Steve and I interviewed Angeline Lo's former manager, whom she highly recommended. As soon as the guy walked into the room, I knew it was going to be problematic, because he seemed extremely straight-laced and uptight, dressing more like an insurance salesman than a technologist. He also seemed very nervous as he fumbled at our first few questions.

I could tell that Steve was losing patience when he started to roll his eyes at the candidate's responses. Steve began to grill him with some unconventional questions.

"How old were you when you lost your virginity?", Steve asked

The candidate wasn't sure if he heard correctly. "What did you say?"

Steve repeated the question, changing it slightly. "Are you a virgin?". Burrell and I started to laugh, as the candidate became more disconcerted. He didn't know how to respond.

Steve changed the subject. "How many times have you taken LSD?"

The poor guy was turning varying shades of red, so I tried to change the subject and asked a straight-forward technical question. But when he started to give a long-winded response, Steve got impatient again.

"Gooble, gobble, gobble, gobble", Steve started making turkey noises. This was too much for Burrell and myself, and we all started cracking up. "Gobble, gobble, gobble", Steve continued, laughing himself now.

At this point, the candidate stood up. "I guess I'm not the right guy for this job", he said.

"I guess you're not", Steve responded. "I think this interview is over."

# And Another Thing...

**Author:** Andy Hertzfeld
**Date:** March 1982
**Characters:** Rich Page, Steve Jobs, Bill Atkinson, Burrell Smith, Andy Hertzfeld, Larry Tesler
**Topics:** Lisa, Management, Personality, Personality Clashes, Lisa Rivalry
**Summary:** There was some friction between the Mac and Lisa teams
**Revision:** most recent of 4

By early 1982, the Macintosh was beginning to be acknowledged as a significant project within Apple, instead of a quirky research effort, but it still remained somewhat controversial. Since the Mac was sort of like a Lisa that was priced like an Apple II, it was seen as potential competition from both groups. Also, our leader Steve Jobs had a habit of constantly boasting about the superiority of the Mac team, which tended to alienate everybody else.

Larry Tesler, who came to Apple from Xerox PARC in the summer of 1980, was the manager of the Lisa Application Software team. He understood and appreciated the potential of the Macintosh and was very supportive of the project. He was concerned that some of the Lisa team didn't share his enthusiasm, and thought that it would be helpful for us to demonstrate the Mac to his team and talk about our plans with them. He arranged for Burrell Smith and me to give a demo during a lunch-time meeting.

By this point, we had stand-alone Macintosh prototypes that no longer depended on an umbilical cord to a hosting Lisa. We didn't have the real plastic cases yet, but we were able to house the prototypes in plastic boxes of around the same size that were a passable imitation. The demo software environment was based on the "Lisa Monitor", a simple operating system cooked up by one of the main Lisa architects, Rich Page, that I got running on the Macintosh. The monitor was based on the UCSD Pascal system Filer and offered a simple, menu-based UI. We were able to boot the Mac into the monitor from an Apple II floppy, and then use it to launch various demo programs.

Burrell and I set up the prototype in a large conference room in the Lisa building. The Lisa applications team was seated around the table, but quite a few other Lisa team members had also gathered around, standing room only, perhaps twenty-five people in all. Larry Tesler gave us a nice introduction, and then we booted up the prototype and started to run through various demos while explaining the capabilities of the machine. Everything seemed to be going well, when suddenly there was a loud, insistent knock at the conference room door.

Before anyone could respond, the door was flung open, and in strode Rich Page, the systems wizard who was one of the main designers of the Lisa. Rich was a tall, bearded, ursine engineer, equally adept at hardware and software, who was responsible for getting Lisa to use the 68000, and had personally ported or created many of the tools that both the Macintosh and Lisa teams were using. But I had never seen him looking as angry as he was at the moment.

"You guys don't know what you're doing!", he began to growl, obviously in an emotional state of mind, "The Macintosh is going to destroy the Lisa! The Macintosh is going to ruin Apple!!!"

Burrell and I didn't know how to respond, and neither did anyone else in the room. Larry Tesler gave me an embarrassed glance, trying to figure out what to do. But Rich wasn't particularly interested in a response, he just wanted to vent his frustration.

"Steve Jobs wants to destroy Lisa because we wouldn't let him control it", Rich continued, almost looking like he was going to start crying. "Sure, it's easy to throw a prototype together, but it's hard to ship a real product. You guys don't understand what you're getting into. The Mac can't run Lisa software, the Lisa can't run Mac software. You don't even care. Nobody's going to buy a Lisa because they know the Mac is coming! But you don't care!"

With that, he turned around and strode out of the conference room, as quickly as he had come in. He slammed the door as he left, with the noise reverberating ominously in the stunned silence. There was some nervous laughter, but nobody knew what to say. Larry Tesler started to apologize, explaining that Rich didn't speak for most of the Lisa team, when suddenly the door was flung open again and Rich Page was back, just as angry as before.

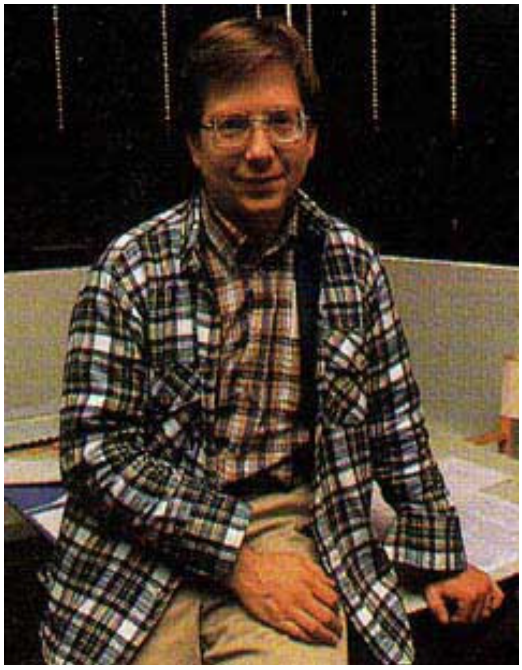
"And another thing...", he said, before pausing to look directly at Burrell and myself. "I don't have any problem with you, I know it's not your fault. Steve Jobs is the problem. Tell Steve that I think that he's destroying Apple!" Once again, he turned around and left abruptly, slamming the door for a second time. We steeled our ourselves, wondering if he was going to return for a third round.

But this time, Larry was able to finish apologizing, and then we finished the demo quickly and held a brief question and answer session, with everyone still a bit shell-shocked from the unexpected outburst. We told Steve Jobs about Rich Page's oration later that afternoon, and he just shrugged, "That's Rich Page for you. He'll get over it."

The next morning Bill Atkinson called and told me that Rich Page felt bad about what happened and he wanted to take Burrell and me out to lunch to apologize. So that afternoon, the four of us went out for a long lunch, where Bill explained that Rich was just trying to do what he thought was right, and he didn't intend to get so emotional. Rich told us that he really appreciates that Burrell and I were doing great work for the company, but he was frustrated that Steve was such a loose cannon, and wasn't working for our mutual success. We left on decent terms, but in the back of my mind I was still worried that such obvious resentment would be a problem for us in the future.

# And Then He Discovered Loops!

**Author:** Andy Hertzfeld
**Date:** April 1982
**Characters:** Bob Belleville, Rich Williams
**Topics:** Software Design, Management
**Summary:** Bob has written many lines of code
**Revision:** most recent of 3



**Bob Belleville in 1983**

We interviewed quite a few candidates to replace Bud as the software manager before encountering Bob Belleville, who was one of the main hardware designers of the Xerox Star, the first commercial computer with a graphical user interface. He was intelligent and soft-spoken, and dryly skeptical about human nature. One of his many aphorisms was "The Law of Conservation of Misery" (no matter what course of action is taken, the total human misery in any given situation is maintained), which seemed particularly applicable to large computer companies.

Bob's background looked to be a lot stronger in hardware than software, so we were somewhat skeptical about his software expertise, but he claimed to be equally adept at both. His latest project was a rebellious, skunk-works type effort to make a low cost version of the Star called "Cub" that used an ordinary Intel microprocessor (the 8086), which was heresy to the PARC orthodoxy, who felt that you needed custom, bit-slice processors to get sufficient performance for a Star-type machine.
Bob had written much of the software for Cub himself.
"I've got lots of software experience", he declared, "in fact I've personally written over 350,000 lines of code."

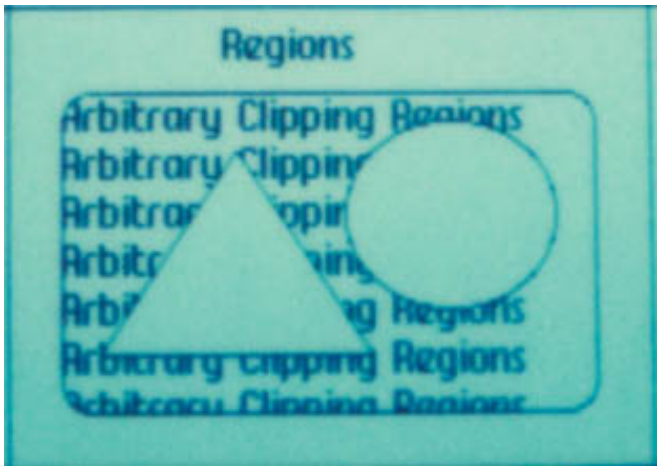
I thought that was pretty impressive, although I wondered how it was calculated. I couldn't begin to honestly estimate how much code I have written, since there are too many different ways to construe things.

That evening, I went out to dinner with my friend Rich Williams, who started at Apple around the same time that I did. Rich had a great sense of humor. I told him about the interview that I did in the afternoon, and how Bob Belleville claimed to have written over 350,000 lines of code.

"Well, I bet he did", said Rich, "but then he discovered loops!"

# I Still Remember Regions

**Author:** Andy Hertzfeld
**Date:** April 1982
**Characters:** Bill Atkinson, Steve Jobs
**Topics:** Software Design, QuickDraw, Personality
**Summary:** We almost lose Bill in a car accident
**Revision:** most recent of 5




**Regions were the heart of QuickDraw**

The single most significant component of the original Macintosh technology was QuickDraw, the graphics package written by Bill Atkinson for the Lisa project, which pushed pixels around the frame buffer at blinding speeds to create the celebrated user interface. One of QuickDraw's main jobs was to provide the primitives for quickly drawing text and graphics into overlapping windows, when the window that you're drawing into may be partially obscured by other windows. Applications could just draw without worrying if their window was obstructed because Quickdraw, with a little help from the window manager, would take care of the clipping to make sure pixels stayed inside in their window.

Overlapping windows can form complex shapes, especially if their corners are rounded. The key data structure in QuickDraw was called a "region", which compactly represented an area of the screen. QuickDraw provided routines that allowed the programmer to define regions by using the built-in drawing primitives, and to perform operations with them like union and intersection. Most importantly, all of the QuickDraw drawing primitives clipped to the intersection of three different regions, to allow drawing into obscured windows. We considered QuickDraw's speed and deftness at region handling to be the most significant "crown jewel" in Apple's entire arsenal.

The region data structure was a variable-sized list of what Bill called "inversion points", the coordinate values where black changed to white or vice versa. Since most regions were mostly rectangular, there weren't many inversion points, so regions were quite compact. But occasionally, there were lots of inversions, like in a circle, so regions grew as necessary.

QuickDraw was written entirely by Bill Atkinson, and in the spring of 1982 it was still evolving. He had recently sped up region operations by more than a factor of four. The concept of "pictures", a set of drawing operations grouped together for easy playback, was just added to the package, and hadn't really settled down yet. At this point, the Lisa applications were beginning to come together and Bill was changing QuickDraw in response to what they needed.

One morning, we were shocked to hear that Bill had gotten into a really bad car accident on his way into work. Apparently, he had turned a corner and not seen a parked truck, and slammed his little Corvette into the truck, shearing the roof off the top of his car. Bill was knocked unconscious and got pretty banged up, although he was still in one piece - one of the police officers who surveyed the wrecked Corvette commented that it was a miracle that Bill wasn't decapitated. This was a little more than a year after Woz's plane crash, but it brought back memories of that.

When Steve Jobs heard about the accident, he immediately jumped into his car and drove to the hospital where they had taken Bill. He was in a hospital bed, and had only recently regained consciousness by the time Steve arrived. He sustained a head injury and lost some blood, but luckily there were no major problems.

Steve entered the hospital room and was relieved to see that Bill had regained consciousness. "Is everything OK?", Steve asked. "We were pretty worried about you."

Bill turned his head and looked at Steve. He managed a painful smile. "Don't worry, Steve, I still remember regions."

# You Can't Fire Bruce!

**Author:** Andy Hertzfeld
**Date:** May 1982
**Characters:** Bob Belleville, Bud Tribble, Alan Kay, Steve Jobs, Bruce Horn, Ted Kaehler
**Topics:** Management, Personality Clashes
**Summary:** The software team has a personality clash
**Revision:** most recent of 12

Even though he was only twenty two years old when he joined the Mac team, Bruce Horn already had seven years of experience tinkering with graphical user interfaces. He was recruited as a 14-year-old by Ted Kaehler to do some programming experiments in Smalltalk at Alan Kay's Learning Research Group in the mid-seventies, and he took to it so well that he had worked part-time at Xerox PARC ever since. By the time he joined the Mac team in late 1981, he was an expert in object-oriented programming and graphical user interfaces.

Bruce was a bright, idealistic and uncompromising programmer who fit right in with the prevailing values of the team, and he quickly became an important contributor to the Macintosh system software effort. He was slated to work on the application that graphically represented files, which Bud Tribble had dubbed the "Finder", but after a few weeks he convinced us that we were missing a crucial level of the system that he called the "Resource Manager", which was a way to manage the various chunks of data (like text strings and images) that an application relied on, and which was going to be needed for the Finder application as well for managing icons and bindings between documents and applications (see Resource Manager Countdown) .

Bruce was busy implementing the resource manager when Bob Belleville arrived on the scene as the new software manager in April 1982, replacing Bud Tribble, who had to return to medical school at the end of 1981. Bob was also a Xerox alumnus, but otherwise they couldn't have been more different. Bob was pragmatic and somewhat authoritarian, with his world-view substantially forged by his stint in the Navy, and he immediately began to clash with Bruce's idealism.

I spent a lot of time with Bob when he first started on the project, helping him to get up to speed, and we got along well at first, even though we had very different perspectives. Bob was a little bit older than most of the core team, with a wife and family, and he was naturally skeptical. He didn't understand why we were so excited about the Macintosh, which he saw as just another increment along a continuum. "I don't get it", he complained to me, "This computer will be obsolete in a year, and then there will be another one, and another one after that. What's so special about this one?"

The first software team meeting that Bob presided over took place in the main conference room of our new building, soon after we moved from Texaco Towers to Bandley 4, around a long table that Burrell and I used for afterhours ping-pong. Up to this point, the software team had a very loose management style, without formal meetings, so this was something new for us. Bob had us go around the table, telling the group what each individual was currently working on, and when we were planning to complete it.

When it was Bruce Horn's turn, he described his Resource Manager work, but refused to give a date as to when it would be finished. Bob visibly bristled, and asked him to make the best estimate that he could, which Bruce refused to do, claiming that it wouldn't be accurate. Bob wanted Bruce to put the resource manager aside to work on the Finder, which Bob thought had much higher priority, but Bruce refused to go along with him, because of the dependence of the Finder design

on the resource manager. They were at an impasse, but finally Bob said he'd resolve it by talking with Bruce privately, and the meeting was able to move on.

I was worried about how the obvious tension between Bruce and Bob would be resolved. A few days later, Bruce came up to me in the late afternoon, visibly shaken.

"You wouldn't believe what just happened! I finally had my meeting with Bob about the resource manager."

"So what happened?"

"We started off talking about the work that was needed to finish it, but I guess I said something that he didn't like. He started getting all weird, and told me that I was insubordinate, and that he was my manager, and that I had to follow his orders or else. Or else what?, I asked him. You wouldn't believe what he did!"

"What did he do?", I asked him, not knowing what to expect.

"He threatened me! Can you believe that? For a second I thought we were going to have a fist fight, but he started smiling as if he was joking. I didn't know what to think. Finally, he said we'd talk more later and walked away."

The image of Bruce and Bob duking it out was ludicrous, as Bruce was at least 8 inches taller than Bob and probably more than 60 pounds heavier. Bruce's father was an ex-professional football player before he became a doctor, and Bruce was built the same way. I tried to calm him down and told him that I would help him get this resolved. The next morning I went to Bob Belleville's office as soon as I arrived at work.

"Good. I'm glad that you're here", Bob greeted me. "I need to talk to you about Bruce Horn."

"Yes, that's what I want to talk about", I responded. "Bruce told me about the conversation that you had last night."

"Well, it's not a problem anymore", Bob said, flashing a cryptic smile.

"Why not?" Something didn't sound right to me.

"I decided to fire him. He doesn't respect authority enough to work on this team."

Now it was my turn to get upset. "You can't fire Bruce!", stating something that I thought was obvious. "He's doing really important work for us, and losing him would set us back months."

"Keeping him will set us back even more, because he's a trouble maker, and he'll cause more trouble in the future."

But I was adamant, defending Bruce until Bob got frustrated with me. "Well, Andy, I am very disappointed in you. I thought you had more sense than that." I walked out of his office not knowing what was going to happen.

Later that day, when Steve Jobs came by for his usual early evening visit, I told him what transpired and repeated how bad it would be if we lost Bruce. I don't know what Steve eventually said to Bob, but he apparently dropped his plan to fire Bruce since he never mentioned it again. But I never thought of Bob the same way after that, and I think that was when he started to have problems with me as well.

# Bouncing Balls

**Author:** Caroline Rose
**Date:** May 1982
**Characters:** Chris Espinosa, Caroline Rose
**Topics:** Recruiting
**Summary:** My interview with Chris Espinosa
**Revision:** most recent of 7

**Caroline Rose in 1982**

When I interviewed for a job in the Macintosh group in May 1982, I was working at Tymshare (just down the block), where I had been immersed in the computer timesharing industry for so long that I hadn't kept up with what was going on in the world of personal computing. But I'd always been quick to learn new technologies, so I wasn't worried when I went to talk to Chris Espinosa about the possibility of writing what would come to be known as Inside Macintosh.

I should add that at Tymshare, in a group acquired from SRI, I'd worked for a while under Doug Engelbart and used a rather complicated mouse device he'd invented. It had struck me as an interesting experiment but a bit gimmicky. I'd also played Hangman on a computer at Xerox PARC with a friend who worked there; the graphics seemed pretty cool, but I thought of it only as a game/toy — otherwise why wouldn't Xerox have been attempting to market it for more serious purposes? Much earlier I had used electronic messaging on the ARPAnet, mainly to communicate with fellow workers in the building, and at first thought it a silly substitute for walking down the hall to talk to someone in person. So you can see that the light about the potential for such innovations dawned very slowly on me.

At Apple, the interview with Chris went well. He was very enthusiastic about my skills, and of course about what the Mac group was doing. He demonstrated a prototype of the menu command interface on the Mac, but was even more excited to show me a demo of balls bouncing all over the screen, as yet another indication of how this little computer would "change the world." I guess he thought a picture was worth a thousand words, because he didn't fill in a lot of the blanks for me: it certainly seemed cute enough, great for games and such, but I couldn't see much beyond that.

Later that day, back at Tymshare, a programmer friend of mine asked me how the interview had gone. When I got to the part about the balls bouncing on the screen, my friend became tremendously excited. He pushed me into a conference room, closed the door, and proceeded to point out the significance of what I'd seen. I don't recall the details, but I remember that he opened my eyes to see beyond Hangman in terms of what graphics — and especially animated graphics — might come in handy for on a computer. I felt like I was looking into a crystal ball; I did, finally, get the picture. (And, of course, I got the job!)

# Do It

**Author:** Andy Hertzfeld
**Date:** June 1982
**Characters:** Larry Tesler
**Topics:** User Interface, Software Design, Lisa
**Summary:** User testing sometimes has surprising results
**Revision:** most recent of 12

Many of the academic types who were involved in creating the earliest implementations of the graphical user interface at Xerox PARC and various universities sort of sneered at the first generation of personal computers when they appeared in the mid-seventies, since the early personal computers were much less powerful than the machines that they were used to programming. There wasn't that much you could do with only four kilobytes of memory and no disk drive.

But Larry Tesler, who was a key member of the Smalltalk team in the Learning Research Group at Xerox PARC, felt differently. He was really excited by the potential of personal computers, buying a Commodore PET as soon as one became available in 1977. He was one of the demonstrators at Apple's famous Xerox PARC visit in December 1979, and he was so impressed by the Apple visitors that he quit PARC and started working at Apple on July 17, 1980, as the manager of the Lisa Applications team.

Larry championed consistency between applications, and made many contributions to what eventually became the Macintosh User Interface. He was also the leading advocate and implementor at Apple of user testing: actually trying out our software out on real users and seeing what happened. Starting in the summer of 1981, Larry organized a series of user tests of the nascent Lisa software, recruiting friends and family to try out the software for the first time, while being observed by the Apple designers who recorded their reactions.

The user tests were conducted in a specially constructed room featuring a one-way mirror, so observers could watch the tests without being intrusive. The tests were conducted by a moderator who made sure the user felt comfortable and showed her the basics of using a mouse. Then, with no further instruction, users were asked to perform specific tasks, without help from the moderator, like editing some text and saving it. The moderator encouraged each user to mumble under her breath while doing the tasks, revealing her current thinking as much as possible. Each session was audio or videotaped for later analysis.

When the software required confirmation from the user, it displayed a small window called a "dialog box", that contained a question, and presented two buttons, for positive or negative confirmation. The buttons were labeled "Do It" and "Cancel". The designers observed that a few users seemed to stumble at the point that the dialog was displayed, clicking "Cancel" when they should have clicked "Do It", but it wasn't clear what they were having trouble with.

Finally, the team noticed one user that was particularly flummoxed by the dialog box, who even seemed to be getting a bit angry. The moderator interrupted the test and asked him what the problem was. He replied, "I'm not a dolt, why is the software calling me a dolt?"

It turns out he wasn't noticing the space between the 'o' and the 'I' in 'Do It'; in the sans-serif system font we were using, a capital 'I' looked very much like a lower case 'l', so he was reading 'Do It' as 'Dolt' and was therefore kind of offended.

After a bit of consideration, we switched the positive confirmation button label to 'OK' (which was initially avoided, because we thought it was too colloquial), and from that point on people seemed to have fewer problems.

# Invasion of Texaco Towers

**Author:** Jerry Manock
**Date:** June 1982
**Topics:** Celebrities, Industrial Design, Social Life
**Summary:** A double standard for secrecy
**Revision:** most recent of 5

**Joan Baez**

The Macintosh Team was sworn to the utmost secrecy about our project. We were moved to the top floor of a non-descript two story office building about two blocks from the established Bandley Drive Apple complex. There was no identification on our door. Our view west was of a Texaco gas station... thus the name "Texaco Towers" spontaneously evolved for the new Macintosh-in-development headquarters (see Texaco Towers) . Steve Jobs would come over to visit us several times a day to stay on top of our progress. On these visits he alternated between "cheerleader" and "strict parent." Ever-present was his enthusiasm, his dedication to excellent design, and his exhortation to keep our project confidential.

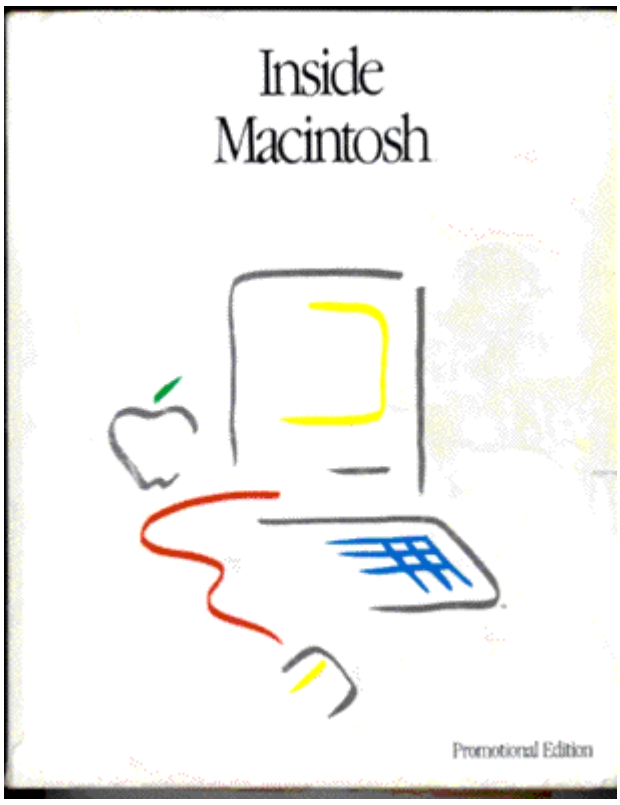
The Jef Raskin Macintosh hardware concept was of a "portable" computer with a keyboard that rotated up to cover and protect a small rectangular CRT screen next to the floppy disk drive. One day Steve came bounding through our Texaco Towers door to announce that the overriding theme was now "minimal footprint on a desk" instead of "portability." He had been to a mall over the weekend and had been looking at "appliances." This term was to have considerable marketing usage in the next few years. Terry Oyama and I immediately started sketching a design that had the CRT above the floppy disk drive and motherboard, which gave the desired smaller footprint. To avoid a boxy look with sharp edges we felt would be intimidating to the user, we employed radiuses on the side corners and a large chamfer on the back. Another goal of ours was to make the back of the computer (which we realized could, in fact, be facing a visitor to user's office) as aesthetically pleasing as the front. Terry and I had traveled to the Hanover (Germany) Fair previously and determined that this would be a radical departure from current practice. For a similar reason we always tried to deal with "cable management" in an aesthetic way instead of the common "rat's nest" still seen today behind EDP equipment. But I digress.

One afternoon, when the project was in its advanced stages, Steve burst through the door, unannounced, in an exuberant mood. He had two guests... Joan Baez and her sister, Mimi Farina. Steve had been to lunch with them nearby and apparently could not contain himself when Joan asked him for advice on which computer to buy for her son, Gabe. Not only did he tell them about our Macintosh-in-development but he decided to SHOW it to them too. We sat there doubly dumfounded at the disclosure of our secret project to an outsider... who happened to be a huge celebrity... that we actually got to meet! Hopefully Steve had them sign a non-disclosure agreement, but I never saw it.

This was not the last time we saw Joan Baez. Steve invited her to an Apple Macintosh Black Tie "Christmas" party one February at the St. Francis Hotel in San Francisco. I have a vivid memory of being on the dance floor with my wife, waltzing between dinner courses to the music of the San Francisco Symphony Orchestra, and bumping into Joan and Steve as they went swirling by. Apple sure knew how to throw a party!

# Inside Macintosh

**Author:** Andy Hertzfeld
**Date:** June 1982
**Characters:** Steve Jobs, Joanna Hoffman, Randy Wigginton, Bill Atkinson, Chris Espinosa, Caroline Rose, Steve Chernicoff, Brad Hacker, Bob Anders
**Topics:** Documentation, Software Design, Lisa
**Summary:** Developer documentation was crucial to our success.
**Revision:** most recent of 15



**The phone book edition of Inside Macintosh**

One of the main differences between the Lisa and Macintosh projects was the way that they viewed third party developers. The Lisa team was writing an integrated suite of seven office-oriented applications internally, so they didn't see a need to support third party developers at first, although they planned to do it eventually.

The Macintosh, inspired by the Apple II, took a different approach. The Apple II's sales had increased more than ten-fold when a tiny company named Software Arts released the first spreadsheet, Visicalc, which initially ran only on the Apple II. We wanted all the people who resonated with our Macintosh dream to be able to extend it with own creativity, so having first class support for third party developers was considered to be a must, from the very beginning.

But that was easier said than done. Lisa's stance was quite reasonable, because consistency between applications was very important to us. There were virtually no third party developers who were familiar with a graphical user interface, so we had to educate them about a whole new approach to programming. In those days, every application provided its own unique user interface, and we weren't sure that it was even possible to coordinate independent developers to conform to our ideas about a consistent user interface.

In fact, in early 1982 our user interface was still evolving, and everyone on the team didn't necessarily agree about the best way to do things, especially in situations that hadn't been considered yet. It seemed like the next logical step was to formally document and codify our user interface, to identify and resolve open issues and communicate it to third party developers.

One forcing function was a looming meeting with our initial third-party developer, Microsoft (see Shut Up!) , who was coming to pick up their first Macintosh prototypes and developer documentation around the end of January. We held a series of intense, all day meetings in the second week of January 1982, to thrash out disagreements and formulate a shared view of the UI, metaphorically locking ourselves in a room until we came to consensus.

The meetings were attended by Steve Jobs, Bill Atkinson, Joanna Hoffman, Chris Espinosa, Randy Wigginton (who had left Apple in September 1981, but agreed to write MacWrite for us as a semi-independent developer) and me.

The Lisa User Interface was our obvious starting point, but we had a drive to simplify things, and tried to strip out anything that seemed too complex. Triple-click was easy to get rid of, for example, but it was hard to agree about the details of scroll bars. After two and a half days, we thought that we were more or less in agreement, and decided that Joanna should write up the current state of the design. She wrote the first draft of the "Macintosh User Interface Guidelines" in time for the Microsoft meeting the following week, where we presented it for the very first time. Eventually, Chris Espinosa took over authorship of the guidelines, augmenting and amending them as necessary as development proceeded.

By April 1982, the first implementation of the User Interface Toolbox, which contained the code that implemented UI objects like windows, menus, buttons and scrollbars, was ready for the initial release to developers. Since most developers had never programmed a graphical user interface before, it was very important to write high quality developer documentation to explain the ins and outs of using the toolbox. Chris Espinosa had already written some excellent documentation for using QuickDraw, so we were off to a good start.

I met with Chris about the toolbox documentation and we decided to tackle the window manager first. He assigned a recently hired technical writer, a women in her mid-twenties who will remain mercifully nameless in this story, to work with me to document the Window Manager API.

One afternoon, I sat down with the writer for a few hours with some print-outs of the source code, and went over the Window Manager API with her in fine detail. I was a bit worried because I did most of the talking, and she didn't ask any questions, but she promised to show me her first draft in just a few days.

A few days later, Chris Espinosa handed me a few pages of Window Manager documentation, with the caveat that it was a very early draft, so I shouldn't expect too much. But my heart sank as I began to read it. The actual window manager calls were accurately reproduced, as were the comments from the header file, but the descriptions of each call made no sense; it was clear that she didn't understand many of the underlying QuickDraw and memory manager concepts, and instead of asking for an explanation, she just made up whatever popped into her head.

I had a panicky meeting with Chris, but he was able to calm me down and convince me to give the writer another chance. I met with both of them, explaining the problems that I had with what she had done so far. She was amazingly blithe and cheerful about it, saying that she knew that she didn't understand everything, but figured that I would correct anything that was wrong. We had another longer meeting, where I did my best to explain the underlying concepts like handles and regions, and went over the Window Manager API again, this time asking her if she had any questions at the end of each routine. It still seemed to me like she was having trouble understanding things, no matter how carefully I tried to explain, but she didn't seem worried about it at all.

The next draft was just as bad as the previous one, and I felt even worse, given all the effort that I put into it. Chris was defensive, and I began to despair of ever getting decent documentation for the toolbox. So I was suprised when he entered my cubicle a couple of days later, with a smile on his face.

"We've just made an offer to a new writer", he told, "someone who I think will do a much better job on the technical side of things, since she used to be a programmer. Her name is Caroline Rose. I'm going to assign her to the window manager documentation and see what you think."

The next week I sat down to meet with Caroline for the first time, and she couldn't have been more different than the previous writer. As soon as I began to explain the first routine, she started bombarding me with questions. She didn't mind admitting it when she didn't understand something, and she wouldn't stop badgering me until she comprehended every nuance. She began to ask me questions that I didn't know the answers to, like what happened when certain parameters were invalid. I had to keep the source code open on the screen of my Lisa when I met with her, so I could figure out the answers to her questions while she was there.

Pretty soon, I figured out that if Caroline had trouble understanding something, it probably meant that the design was flawed. On a number of occasions, I told her to come back tomorrow after she asked a penetrating question, and revised the API to fix the flaw that she had pointed out. I began to imagine her questions when I was coding something new, which made me work harder to get things clearer before I went over them with her.

Initially, we distributed the raw documentation to developers piecemeal, as it was written, but eventually we wanted to collect it into one definitive reference called "Inside Macintosh". It was almost 1000 pages long, spread across three volumes, mostly written by Caroline with help from Bob Anders, Brad Hacker, Steve Chernicoff and a few others. Steve Jobs insisted on very high production standards for the first edition, naturally, using only the best binding and paper available. But high quality printing takes time, and the evangelists were impatient to get the definitive documentation out to developers as soon as possible.

I'm not sure whose idea it was, but a compromise was finally reached. Apple would publish a free, soft-bound "promotional" edition of Inside Macintosh on low quality paper as soon as possible, and send a copy for free to every developer. It was about as thick and flimsy as the Yellow Pages, so it became known as the "phone book" edition. Most developers still bought the high quality, beautiful hardback edition when it came out a few months later, anyway.

# Alice

**Author:** Andy Hertzfeld
**Date:** June 1982
**Characters:** Steve Capps, Bruce Daniels, Steve Jobs, Joanna Hoffman
**Topics:** Software Design, Games, Lisa, Marketing
**Summary:** The Macintosh's first great game
**Revision:** most recent of 15

**Alice's initial screen**

Even though Bruce Daniels was the manager of the Lisa software team, he was very supportive of the Mac project. He had written the mouse-based text editor that we were using on the Lisa to write all of our code, and he even transferred to the Mac team as a mere programmer for a short while in the fall of 1981, before deciding that he preferred managing for Lisa. He would sometimes visit us to see what was new, but this time he had something exciting to show us.

"You've got to see the new game that Steve Capps wrote", he told me while he was connecting his hard drive up to my Lisa. He booted up into the Lisa Monitor development system, which featured a character-based UI similar to UCSD Pascal, and launched a program named "Alice". Steve Capps was the second member of the Lisa printing team, who started at Apple in September 1981. I had seen him around but not really met him yet.

The screen turned black and then, after a few seconds delay, a three dimensional chess board in exaggerated perspective filled most of the screen. On the rear side of the board was a set of small, white chess pieces, in their standard positions. Suddenly, pieces started jumping into the air, in long, slow parabolic arcs, growing larger as they got closer.

Soon there was one specimen of each type of piece, all rather humanoid looking except for the tower-like rook, lined up on the front rank of the board, waiting for the player to click on one to start the game. The player would be able to move like the piece they chose, so it was prudent to click on the queen, at least at first.

The pieces jumped back to their natural positions on the far side of the board, and an image of a young girl in an old fashioned dress floated down to the front row, which represented Alice from Lewis Carroll's "Alice in Wonderland" and "Through the Looking Glass" books, drawn in the style of the classic John Tenniel illustrations. The player controlled Alice and viewed the board from her perspective, facing away from the player so you only saw her from the rear.

A three digit score, rendered in a large, ornate, gothic font, appeared centered near the top of the screen, and then the game began in earnest, with opposing chess pieces suddenly leaping into the air, one at a time, in rapid succession. If you stood in one place for too long, an enemy piece would leap onto Alice's square, capturing her and ending the game.

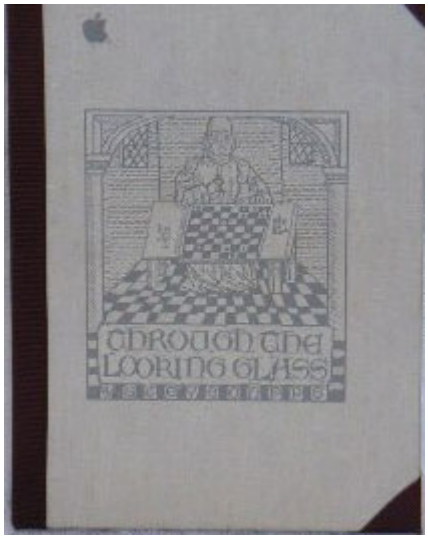
It didn't take long to figure out that if you clicked on a square that was a legal move, Alice would leap to it, so it wasn't too hard to jump out of the way of an enemy piece. And, if you managed to leap onto the square of another piece before it could move out of the way, you knocked it out of action and were rewarded with some points. You won the game if Alice was the last one standing.

I was impressed at the prodigious creativity required to recast "Through The Looking Glass" as an action-packed video game that was beautiful to behold and fun to play. Alice was also addicting, although it took some practice to be able to survive for more than a few minutes. Obviously, we needed to have it running on the Mac as soon as possible.

Bruce Daniels seemed pleased that we liked the game. "Capps could probably port Alice to the Mac", he said, anticipating what we were thinking. "Do you think you could get him a prototype?"

Everyone agreed that we should get Capps a Mac prototype right away. I accompanied Bruce Daniels back to the Lisa building (where the rules required that non-Lisa employees be escorted by a Lisa team member), and I finally got to meet Steve Capps, who seemed easy-going and friendly, with a self-deprecating sense of humor. Later that afternoon, he visited Texaco Towers and I gave him the prototype and answered a few questions about the screen address and the development environment. He assured me that it wouldn't take that long to port.

Two days later, Capps came over to present us with a floppy disk containing the newly ported Alice game, now running on the Macintosh. It ran even better on the Mac than the Lisa, since the Mac's faster processor enabled smoother animation. Pretty soon, almost everybody on the software team was playing Alice for hours at a time.




**Alice's packaging**

Within a few weeks, I must have played hundreds of games of Alice, but the most prolific and accomplished player was Joanna Hoffman, the Mac's first marketing person. Joanna liked to come over to the software area toward the end of the day to see what was new, and now she usually ended up playing Alice for longer and longer periods. She had a natural talent for the game, and enjoyed relieving work-related stress by knocking out the rival chess pieces. She complained about the game being too easy, so Capps obliged by tweaking various parameters to keep it challenging for her, which was probably a mistake, since it made the game much too hard for average players.

Steve Jobs didn't play Alice very much, but he was duly impressed by the obvious programming skill it took to create it. "Who is this Capps guy? Why is he working on the Lisa?", he said as soon as he saw the program, mentioning Lisa with a hint of disdain. "We've got to get him onto the Mac team!"

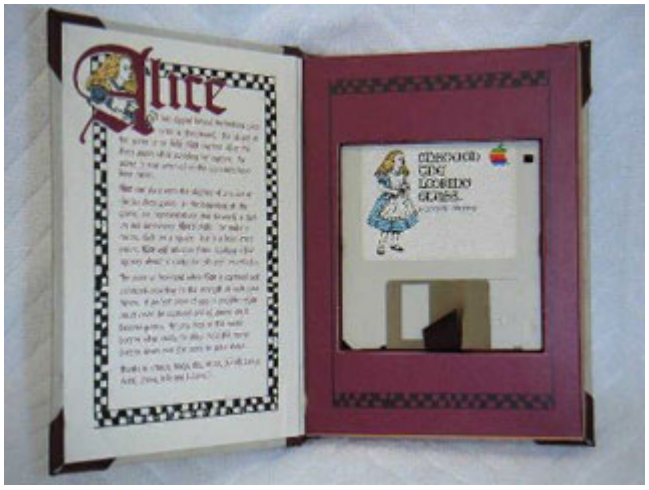
But the Lisa was still months away from shipping, and Capps was needed to finish the printing software, so Steve wasn't able to effect the transfer. One weekend Capps ran into Steve Jobs in Los Gatos and was told, "Don't worry, the Mac team is going to nab you!" Finally, a compromise was reached, that allowed Capps to transfer over in January 1983 after the first release of the Lisa was completed.

Capps quickly became a crucial member of the Mac team, adding fresh energy and talent as we entered the home stretch, helping to finish the Toolbox and the Finder, as well as working on other stuff like "Guided Tour" diskette. But he also found time to embellish Alice with more cool features.

One day he showed me Alice's hidden "Cheshire Cat" menu, which allowed you to adjust various preferences. Alice didn't have a menu bar, so it was sort of part of the game to figure out how to invoke the preferences. It was accomplished by clicking on the score at the top of the screen,

which caused a detailed, John Tenniel inspired Cheshire Cat bitmap to slowly fade into view; clicking on different parts of it set various preferences. Capps also created an exquisite, tiny rendering of the Cheshire Cat to serve as Alice's icon.

Over time, he added some interesting variations, invoked by clicking on various parts of the Cheshire Cat. For example, one variation made some of the squares of the chessboard disappear at random, causing unlucky pieces to fall through to oblivion below. He also added a feature that Woz suggested: as the cursor moved to the back of the board, its image got correspondingly smaller, adding to the illusion of depth.




**Alice's packaging's interior**

By the fall of 1983, Capps started thinking about the best way to get Alice to market. One possibility was publishing it through Electronic Arts, which was founded a year earlier by Trip Hawkins, Lisa's former marketing manager. But Steve Jobs thought that the game at least partially belonged to Apple, and insisted that Apple be the publisher. He negotiated a modest deal with Capps, promising him that Apple would do a deluxe job with the packaging and marketing.

**hidden DK logo**

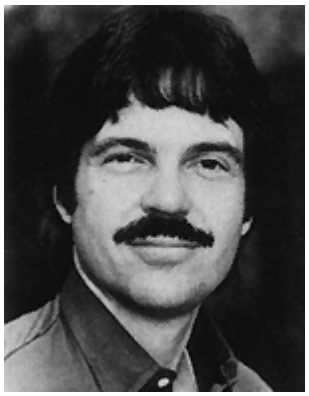
Alice was announced at the launch and featured in the original brochure, but it didn't became available until a couple of months later. True to Steve's word, the packaging was beautiful. The game disk was enclosed in a small cardboard box designed to look like a finely printed, old fashioned book, complete with an elaborate woodcut on the cover, that contained a hidden Dead Kennedy's logo, in tribute to one of Capp's favorite bands. Since Alice didn't take up the whole disk, Capps including a few other goodies with it, including a font and "Amazing", a fascinating maze generating program that he wrote.

When I saw the completed packaging, I was surprised to discover that the game wasn't called "Alice" anymore; apparently, that name was already trademarked for a database program. It was rechristened "Through The Looking Glass" for its commercial release.

Unfortunately, Apple never put the promised marketing effort into Alice. They were in a quandary because the market didn't understand the graphical user interface as a productivity enhancement yet; graphics meant games, so the Mac had to live down an initial reputation as being unsuitable for business tasks. Apple didn't exactly want to promote a game for the Mac at the time, no matter how sensational, so Alice never quite reached as wide an audience as it deserved.

# Creative Think

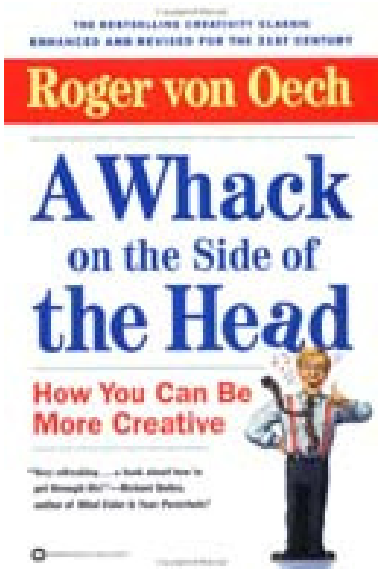
**Author:** Andy Hertzfeld
**Date:** July 1982
**Characters:** Alan Kay, Bill Budge
**Topics:** Inspiration
**Summary:** A seminar with Alan Kay
**Revision:** most recent of 16



**Alan Kay, in the PARC days**

In July of 1982, while I was in the midst of writing the Control Manager part of the Macintosh toolbox, my friend Bill Budge invited me to a computer industry seminar called "Creative Think", where interesting people gave purportedly inspiring talks. It was organized by Roger van Oech, a consultant who had written a book about creativity entitled "A Whack On The Side Of The Head". I usually avoided both creativity seminars and industry schmooze-fests, but my friend Bill had somehow finagled free tickets and I thought it would be worth it just to see him.

Carver Mead gave an interesting talk on newly emerging VLSI technology, and some old chip industry veterans recounted amusing war stories, but the last talk of the day was the one that mattered to me. It was given by Alan Kay, the inventor of Smalltalk and the Alto, and the driving force behind Xerox PARC. I had read about Alan, and been inspired by his article in the September 1977 issue of Scientific American, but I had never seen him before in person or heard much about him.




**Roger's Book**

Alan's speech was revelatory and was perhaps the most inspiring talk that I ever attended. I grew increasingly excited as he made one brilliant, insightful remark after another, and took out my notebook to write as much of it down as I could. Alan was articulating the values behind the work that I was doing, even though he wasn't aware of it, in a way that really resonated with me. After I got back to my office in Cupertino, I transcribed it onto a single page, and copied it to give out to the rest of team.

I still have those notes, so I thought it would be interesting to reproduce them here, as an example of some of the thinking that inspired our efforts.

---

**Alan Kay's talk at Creative Think seminar, July 20, 1982**

Outline of talk: Metaphors, Magnetic Fields, Snobbery and Slogans

The best way to predict the future is to invent it.

Humans like fantasy and sharing:
Fantasy fulfills a need for a simpler, more controllable world.

Sharing is important - we're all communication junkies. We have an incredible bandwidth disparity (easy to take in, hard to give out); our devices have the reciprocal disparity (hard to take in, easy to give out)

Metaphors:
Computer as medium (like clay or paint)
Computer as vehicle
Computer as musical instrument

Magnetic Fields:
Find a central metaphor that's so good that everything aligns to it. Design meetings are no longer necessary, it designs itself. The metaphor should be crisp and fun.

Smalltalk is object-oriented, but it should have been message oriented.

Snobbery:
Turn up your nose at good ideas. You must work on great ideas, not good ones.

Appreciate mundanity: after all, a pencil is high technology

One goal: the computer disappears into the environment

The computer shouldn't act like it knows everything.

The whole notion of 'programming language' is wrong.

Slogans:
Better is the enemy of best
Relative judgements have no place in art
Systems programmers are high priests of a low cult
Point of view is worth 80 IQ points
Good ideas don't often scale

Remember, it's all software, it just depends on when you crystallize it.

People who are really serious about software should make their own hardware.

Final advice: content over form, go for fun.

# You Guys Are In Big Trouble

**Author:** Andy Hertzfeld
**Date:** August 1982
**Characters:** Steve Jobs, Bill Atkinson, Bud Tribble, Andy Hertzfeld, Bruce Horn
**Topics:** Apple Spirit, Buildings, Management
**Summary:** We receive Steve's permission to dismantle the onerous burglar alarm
**Revision:** most recent of 3

By the spring of 1982, the Mac team was growing so rapidly that we had to move from Texaco Towers back to the main part of the Apple campus on Bandley Drive. We moved into Bandley 4, a medium sized building across the street from Apple's main office.

One of the things that I liked about Bandley 4 was that the software team was in the very back of the building, near the parking lot, so we could go out the back door to our cars, or to play basketball, without having to walk all the way around the building. This worked out well for a couple of months, but eventually the facilities group decided to put an alarm on the back door, so you couldn't go in and out freely. They didn't arm the alarm until 5:30pm, but that didn't help me, since I usually didn't go out to dinner until after 7pm, and then usually came back to work another few hours. The alarm became a major annoyance, since it made me walk significantly out of my way a few times every day.

Every once in a while someone would forget that the alarm was there, and walk out the door anyway. The alarm would produce a head-splitting wail, destroying any chance of concentration until a security guard arrived to disarm it, which sometimes could take more than ten minutes.

I complained about the alarm every way that I knew how, but to no avail. About one quarter of the software team worked late, and the alarm was unnecessary while we were there, so I begged the facilities supervisor not to turn it on until after midnight, even offering to arm it ourselves when the last person left. But my pleas fell on deaf ears.

Every couple of months, Bud Tribble would come down from Seattle and visit with us. We'd show him the latest work we were doing so he could make his great suggestions. Late one afternoon, he showed up in the software area, and we all gathered around to demo to him, including Bill Atkinson and Steve Jobs.

Bill had done some neat hi-resolution scans with an improved dithering algorithm, and he wanted to show them to Bud. They were on his hard disk in the Lisa building, so he ran out the back door to get it. It was after 5:30pm, so that set off the alarm, and a horrible, loud, pulsating noise filled the room.

It went on for at least three minutes before Steve yelled out, "Can't someone figure out how to stop that thing?"

I saw a chance to vanquish my nemesis. "Are we allowed to damage it to get it to stop?", I asked him.

"Yes, do anything you want, I don't care if you break it," he replied, holding his hands over his ears. "Just get the damn thing to stop!"

Bruce Horn and I ran over to the nearby hardware lab and picked up every tool we could find. I got a hammer and screwdriver, and proceeded to pound the screwdriver into the center of the alarm, driving a stake through the demon's heart. The screwdriver went all the way through to the other side, but the alarm kept sounding.

Finally, Bruce took over and gave the screwdriver a mighty twist, and the whole thing flew apart into a half dozen pieces and fell to the ground. The horrible noise finally stopped.

At that very moment, a grizzled security guard entered the back door, just in time to see us cheering as the wrecked alarm finally gave up the ghost. He looked at us, our tools of destruction still in our guilty hands, and said, "You guys are in big trouble!!! Who is in charge here? You better show me your badge."

Steve stepped forward and handed the guard his badge. "I'll take responsibility for this", he told him.

The guard scrutinized Steve's badge. He looked at Steve, then back at the badge a few times. Finally, he shrugged his shoulders, picked up the pieces of the broken alarm, and walked away, without saying another word.

We were gleeful that the alarm was gone, but exactly one week later, a new one was put in its place, which stayed there until we moved to Bandley 3 a few months later. I'm not sure why, but not even Steve could get them to set it for a later time.

# Resource Manager Countdown

**Author:** Andy Hertzfeld
**Date:** August 1982
**Characters:** Bruce Horn, Bob Belleville
**Topics:** Software Design, Management
**Summary:** Bob gives Bruce thirty days to finish the Resource Manager

**Bruce used one note for each remaining day**

The Resource Manager was one of the cornerstones of the Macintosh Toolbox. It provided a way to manage chunks of data so they could be easily accessed by the code but be stored and edited independently from it. This was the basis of our localization, for example, since text strings could be stored as resources and translated without having to change the code and rebuild the entire application.

Resources were conceived and implemented by Bruce Horn. They came a little late in the design cycle, and we had to rewrite some other parts of the toolbox to take advantage of them, although that wasn't clear initially. But in the summer of 1982 they were definitely one of the riskier parts of the design.

Bob Belleville, our new software manager, assessed the situation and decided that the resource manager was too big a risk, and tried to convince us to drop it from the design. Bruce went apoplectic, and Bob eventually compromised on giving Bruce 30 days to implement the resource manager. Bruce agreed that if it wasn't ready within 30 days, it would be dropped from the project.

The next day, Bruce came to work with a few pads of yellow sticky notes. He tore off one note at a time, wrote a number on it, and then stuck it on the wall of the double-sized cubicle that I shared with him. Soon the cubicle was encircled with little yellow notes, each with a number from 1 to 30.

"This shows me how many days that I have left", Bruce explained. "I'll tear one down each day, and it will remind me of how much time is remaining."

For the next month, the first thing that Bruce did every morning was dramatically tear off the note with the lowest remaining number. As the days passed, he was making good progress but it wasn't at all clear that he was going to make it.

When there were only three notes left, I began to wonder what was going to happen if Bruce missed the deadline. The resource manager was mostly working now, but he didn't seem to be on a path to closure, as he continued to refine the design.

Finally, there was only one note left on the wall. But when Bruce came in that morning, he had a new pad of yellow notes, and instead of taking the last one down, he wrote another ten numbers and posted them up, before taking down the previously final note. I started to laugh.

"It's overtime", he explained. "I swear I'll really be done in ten more days."

I was waiting for Bob Belleville to confront him, and tell him that he missed the deadline, but Bob wisely saw that the resource manager was far enough along that it could stay. And Bruce really did finish in the next ten days, kind of.

# US Festival

**Author:** Andy Hertzfeld

**Date:** September 1982

**Characters:** Dan Kottke, Jerry Manock, Steve Jobs, Bill Budge, Rod Holt, Burrell Smith, Steve Wozniak, Andy Hertzfeld

**Topics:** Celebrities, Personality, Recreation

**Summary:** We travel to Woz's rock festival

**Revision:** most recent of 21

**Steve Wozniak on stage at the US Festival**

As soon as Steve Jobs took over the Macintosh project in January 1981, he recruited many of the crucial early Apple employees who had worked on the Apple II, including Rod Holt, Jerry Manock, Dan Kottke, Randy Wiggington and Apple co-founder Steve Wozniak.

Woz was enthusiastic about the Macintosh and started hanging out at the Mac team's new office at Texaco Towers, reviewing Burrell's design and learning the 68000 instruction set. But he was only working on the project for a couple of weeks when he crashed his Beechcraft Bonanza airplane while taking off near his home in Scott's Valley, sustaining a serious head injury and knocking out a couple of teeth.

The accident put Woz out of commission for almost two months. While he was recuperating, he had time to rethink his priorities and he decided that he wanted to go back to college to earn his undergraduate degree, for which he needed just one more year of classes, instead of returning to Apple. He enrolled in the engineering program at UC Berkeley, in the summer of 1981, under the assumed name of "Rocky Clark", in honor of his dog Rocky and his new wife, Candi Clark.

A couple of months after the crash, Woz was listening to his favorite radio station KFAT and had an inspiration about putting together a music festival, a "Woodstock West" featuring his favorite progressive country music performers. He realized that while he had the financial wherewithal, he didn't know the first thing about the music industry and filed the idea away, after mentioning it to a few friends.

Later that fall, while he was attending classes at Berkeley, Woz was introduced to a new age entrepreneur named Dr. Peter Ellis. Peter was a former college radical, who had organized a "survival fair" at San Jose State University in the 60s, where he presided over the burial of a Ford Pinto. He hit it off with Woz, and was enthusiastic about Woz's Woodstock West idea. Peter came up with the name "The US Festival" (in reaction to the "me" decade), and threw in other ideas like incorporating a technology fair and featuring a satellite linkup with rock musicians in Moscow. Woz wrote a sizable check to fund a new corporation, Unuson (which stood for "Unite Nations Using Singing Over Network"), to create and produce the US Festival, with Peter as the executive director.

Peter put together a team, and plans began to take shape for a impressive 3 day music festival to be held over Labor Day weekend at Glen Helen park in San Bernardino, around an hour away from Los Angeles. Unuson paid top dollar to hire the foremost rock promoter in the country, Bill Graham, to put together a superlative bill of first class bands, including the Police, Fleetwood Mac, Tom Petty, Santana and many others.

Woz encouraged his Apple friends to come to the US Festival by giving us special gold passes that got us in for free, and permitted us to hang around back stage. Burrell Smith, Bill Budge and myself rented a camper and drove down together, skipping work to leave Friday morning so we could be there by the start of festivities that evening.

**A view of the stage from afar**

After the long drive, we parked the camper a half mile away from the festival site, and walked the dusty trail to the main concert area amid the gathering crowd. We arrived just as the sun was setting and the first band, The Gang of Four, was taking the stage. After their set ended, we made our way to the stage area, to test out our gold passes and see if we could find Woz.

The security guard at the stage door verified our gold passes and let us in, but he seemed to resent doing it. I had never been backstage at a rock concert before. There was a nice spread of cold cuts and beverages in front of a line of trailers that served as dressing rooms for the bands. Lots of people were milling around, including roadies, groupies, bodyguards and even the occasional rock star.

Suddenly I heard a very loud noise, as someone rode into the backstage area on the largest motorcycle I had ever seen. At first I thought it was a Hell's Angel, but it turned out to be Bill Graham in a black leather jacket and sunglasses, scowling as he ordered people around. We didn't feel very welcome and were frequently asked to show our passes; it seemed like the festival staff was resentful that Woz's friends were allowed to be there.

We finally spotted Woz and waved to get his attention. He came over to us, looking happy and excited.

"Do you guys want to introduce a band? Which one? I've got it worked out with Bill Graham so my friends can introduce their favorite band if they want to. We still have plenty of slots left."

I was intrigued, since one of my favorite groups, the Kinks, were scheduled for Saturday afternoon. But getting up on stage in front of hundreds of thousands of people sounded utterly terrifying to me. I declined, but then I noticed that Burrell's eyes had lit up and he was very excited.

"Santana? Can I introduce Santana? That would be so cool..." Carlos Santana was one of Burrell's favorite guitarists.

Woz pulled a piece of paper from his pocket and checked it. "OK, Santana is still open," he told Burrell. "I've got you down for introducing Santana. Meet me here backstage, after Eddie Money finishes tomorrow and I'll show you what to do." Santana was the third act scheduled for Saturday, after Dave Edmunds and Eddie Money.

That evening, when we got back to the camper, Burrell wrote a brilliant short, sweet and humorous introduction of Santana to use the next day, and started to memorize it by reading it aloud. It was all he could talk about until we made our way backstage early on Saturday morning. Burrell waited expectantly until Woz showed up. He read Woz his introduction, which Woz really liked.

Woz led Burrell up onto the stage, trying to find Bill Graham to introduce him to Burrell and tell him about the upcoming Santana intro. But lots of people wanted to talk with Woz, and he got distracted, leaving Burrell alone on stage for a few minutes, just as Bill Graham appeared, looking pissed off as usual.

Bill Graham took one look at Burrell and grimaced. "Who the #$*! are you? What are you doing on the goddamn stage?"

Burrell explained that he was a friend of Woz's and was waiting onstage to introduce Santana. He pulled out his notes for the introduction from his pocket and waved them for Bill Graham to see.

"Sure you are," Bill Graham responded sarcastically. " I'm going to introduce Santana. And you're gonna get the #%$*! off the stage right now, this instant!"

Burrell looked around for Woz but couldn't spot him. He started to argue but stopped short when he saw how furious Bill Graham became.

Bill Graham motioned to one of his ubiquitous bodyguards, a huge guy with long hair and tatoos covering his forearms. "Please escort this gentleman from the premises", he ordered peremptorily, "and don't let him return!" The bodyguard literally picked Burrell up off the ground by the back of his shirt collar and carried him off the stage and then completely out of the backstage area.

That was the last we saw of Burrell for the next six hours and we wondered what had happened to him. Finally, he reappeared just before Tom Petty started the final set of the day. Bill Graham's thug had dragged him outside the festival gates and confiscated his gold pass. Burrell didn't know what to do, but eventually he obtained another gold pass by walking a few miles to the house that Woz had rented for the weekend. Woz was sorry about what had transpired and asked Burrell if he wanted to try again on Sunday to introduce another band, but this time Burrell was wary and quickly declined. In fact, he had had enough of the US Festival and persuaded Bill and me to drive back home early on Sunday morning without attending the last day of shows.

# Boot Beep

**Author:** Andy Hertzfeld
**Date:** September 1982
**Characters:** Charlie Kellner, Andy Hertzfeld, Terry Oyama, Steve Jobs
**Topics:** Software Design, Technical, Personality
**Summary:** How the boot sound evolved
**Revision:** most recent of 8

When you powered up an Apple II, it would make a short beep sound to let you know that it was alive. We thought that the Mac should do something similar, once it passed the diagnostics, sort of like an infant's first cry, letting the world know that you actually made it here.

The 1981 Macintosh just had a square wave sound generator, where the software controlled the frequency by loading a value into the VIA's timer. I wrote a boot sound routine that gradually incremented the frequency at an accelerating pace, so it had a whooping quality to it that was almost humorous. People generally liked it, but we knew that we'd have to do something better for the real product.

In August 1982, the Mac was redesigned with much better sound quality, so we had the possibility of a better boot sound, since we now had 8-bit samples to play with. I started experimenting a little bit, to see if I could come up with something.

Around this time, Charlie Kellner decided to transfer to the Mac group from the Apple II group. Charlie was a brilliant Apple II programmer (who wrote a classic hi-res bowling game for the Apple II before he started work at Apple), who was a multi-talented, meticulous perfectionist. He had designed a music synthesizer for the Apple II, called the alphaSyntauri, as a side project that was the basis of a small start-up company. For some reason, he got bored with the Apple II and wanted to try out working with the Mac team.

Charlie saw me messing around with sounds for a new boot beep and told me that he knew of a simple algorithm that might work pretty well. He asked me to fill the sound buffer with a simple square wave, but then make successive passes on it, averaging adjacent samples until everything reached the same level.

I coded it up and we tried it, and sure enough it had a pleasant, distinctive chiming quality. With a little bit of tweaking, it became the famous sound that the Mac made when it powered up, which lasted until the Mac II, which had even better sound capability, in 1987. For the interest of technical readers, here is a link to the source code for the Boot Beep routine , written in 68000 assembly language.

Charlie was pleased that he was able to make a significant contribution in his first week on the project. Inspired, he asked if he could take a prototype home over the weekend for testing. The next Monday he came into work very excited.

"I knew that something wasn't right!", he exclaimed. "The sound is being completely muffled by the case!. But I know how to fix it."

He had done a series of experiments with the Mac that he had taken home over the weekend, and found that the Macintosh's case was baffling and distorting the sound. He even printed out graphics showing the results of his measurements. Then, after analyzing the data, he drilled a hole about the size of a dime in a strategic place, which caused the measurements to improve dramatically.

He started demoing his modified prototype, showing how the hole improved the sound quality. The difference didn't sound that significant to me, but it definitely was an improvement. He showed it to Terry Oyama, who designed the case, and asked him if he could add the hole.

The next day, Steve Jobs came by in the afternoon and asked to hear Charlie's demo. He listened to the two Macs, and then decreed "There's not enough improvement! There's no way that we're going to put an ugly hole in the case! Just forget about it!"

Charlie was pretty disappointed, and never got very enthusiatic about the Mac after that. A couple of weeks later, he transfered back to the Apple II group, leaving the boot beep as his only legacy.

# Sound By Monday

**Author:** Andy Hertzfeld
**Date:** September 1982
**Characters:** Burrell Smith, Andy Hertzfeld, Steve Jobs
**Topics:** Hardware Design, Software Design, Management, Technical
**Summary:** Steve threatens to remove Mac's sound capability
**Revision:** most recent of 6

**Cartoon from IEEE Spectrum**

When Burrell redesigned the Macintosh digital board in August 1982 after the Integrated Burrell Machine effort fell apart, one of the most significant improvements involved the sound generation capability. All of the extra logic available in the IBM chip allowed Burrell to implement four simultaneous channels of sound, each using a custom wavetable. That was too expensive to keep without the custom chip, but he was able to maintain the fundamental capability, DMA-fetched sound using 8 bits per sample and a clever, pulse width modulated digital to analog converter.

Burrell figured that the Mac could still have the four simultaneous voices that he had envisioned for the Integrated Burrell Machine, only now we'll do most of the work in software instead of hardware. So he promised Steve that the redesign would feature great sound, including a four voice synthesizer. Steve approved spending an extra dollar on an improved amplifier, to better match the improved sound generation capability.

But there were an enormous number of different things to work on in the summer of 1982, including getting the Mac Toolbox into good enough shape for developers to start writing applications. Burrell bugged me about writing a sound driver, and especially his cherished "four voice engine", but I wasn't able to get around to it as a few weeks passed.

Steve Jobs started to get impatient about getting to hear our new sound capability for himself. Finally, he pulled Burrell and myself aside late on a Friday afternoon.

"You told me that the new sound would be really great, right?", he asked Burrell. "Well, if I don't hear great sound coming out of that prototype by Monday morning, we're going to remove the amplifier. "

Then he looked at me. "You've had long enough to get the sound going. I want to hear great sound on Monday, or else." Then he stormed off, leaving Burrell and me to figure out what to do.

"I think he's bluffing", I told Burrell. "But what if he's not?"

Burrell didn't seem very upset, in fact he seemed kind of excited. I think he was pleased that Steve was on our case about the sound, since he really wanted me to write the sound routines as soon as possible anyway. He promised to buy me meals all weekend while we stayed at Apple to get the sound going.

I had already written a basic diagnostic that generated a simple square wave, which certainly didn't meet anyone's criteria for great sound, but at least it told us the hardware was working, and worked out the basic details of controlling the hardware. Burrell wanted me to get to work right away on the four voice synthesizer, since he wasn't sure we could actually make that work.

As usual, Burrell's new design was very clever. The Macintosh was already continuously fetching data from memory to drive the video display, interleaving memory bandwidth between the display and processor in a similar fashion to the Apple II. But every 44 microseconds, there was a "horizontal blanking interval" where no video data was needed, so Burrell used that time to fetch data for the sound. That gave us a sample rate of 22kHz, which would allow us to do frequencies up to 11kHz, which isn't too bad.

The sound driver worked at the interrupt level, so sound generation could proceed in the background while the processor performed some other task. It arranged to receive control at the beginning of the vertical blanking interval, which occured every 16 milliseconds. It needed to generate all the sound data for the next 16 milliseconds, which worked out to 44 microseconds for every successive sample. If our calculation took 22 microseconds per sample, for example, the sound generation would be soaking up half of the available processor cycles.

It only took a few hours to write a driver with a simple sound generation loop. It could do two voices fine, but it didn't run nearly fast enough to do four - it took too long to generate each sample, which caused audible glitches as well as making everything else run like molasses. Burrell took a look at my code, and saw that I was using some memory locations during the sound calculation.

"Memory? Are you kidding? You can't hit main memory, you'll never make it that way. You've got to do everything in the registers!"

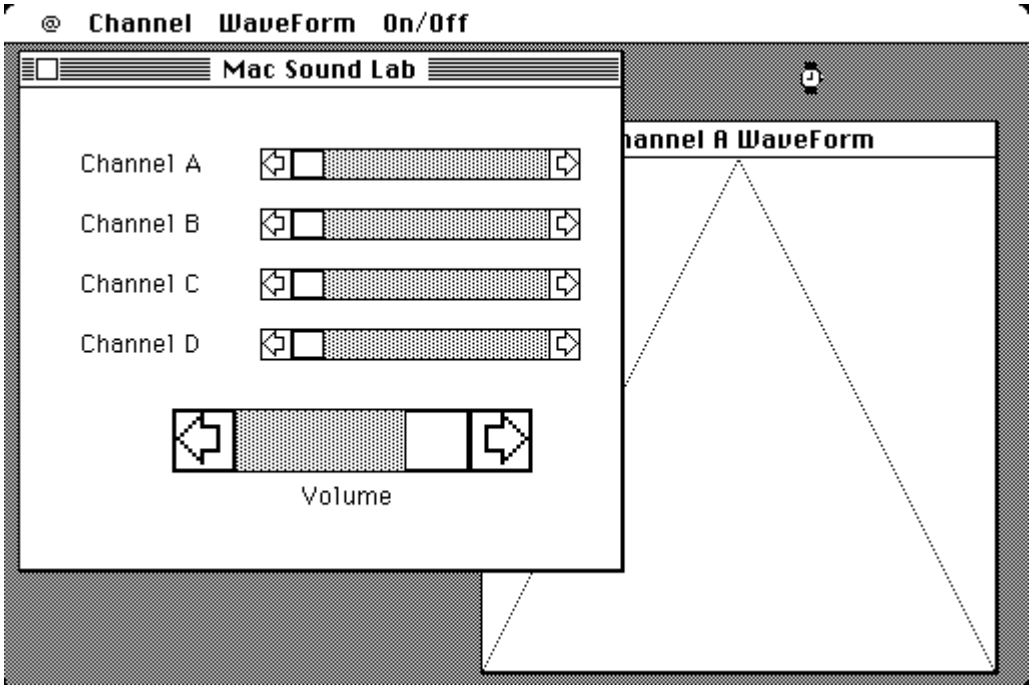
"Registers" are special locations that are part of the processor chip itself, where the action really happens. They could be accessed four times faster than the bulk of memory, which was in separate memory chips. The problem was that there were only 16 registers. For each voice of sound, we needed a frequency, a waveform pointer, a position within the waveform, and an amplitude, plus some housekeeping data. There weren't enough registers to do four voices.

I was able to rewrite the routine without touching main memory, but I was only able to get three voices since I ran out of registers. But that wasn't good enough for Burrell. By now it was late on Saturday evening, and I wanted to go home, but he felt that we had to get the fourth voice done before "head hits pillow" as he liked to say, or we'll never get the fourth voice.

Finally, I was able to leverage the fact that the registers were 32 bits long and we only were doing 16 bit calculations in some of them, to use some of the them to hold two different values. Each sample took about 22 microseconds to calculate, so we were using roughly half of the CPU to get the four voices at the maximum sample rate. The basic four voice capability was implemented, but we still needed an impressive demo to show it off. We went home to sleep at around midnight, after agreeing to come back around noon to work on the demo.

The next day, we decided to write a demo called "SoundLab" that would let the user control the pitch and waveform of the four independent voices. You could specify or edit a waveform by drawing it with the mouse, and control the frequency of each voice with a scrollbar.

The results didn't sound like music, because there was no envelope shaping, but you could make very eerie sounding noises, which we deemed impressive enough. And it was fun to be able to hook an oscilloscope up to the sound output, and then draw a waveform with the mouse and see it on the scope.

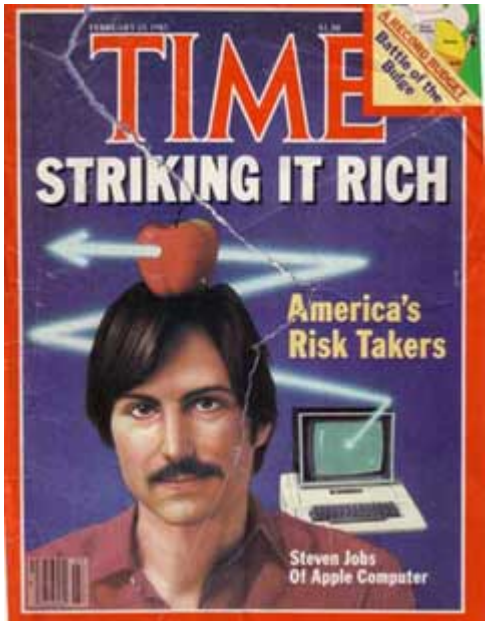



**SoundLab screen dump**

When Steve came in on Monday, he was pleased that we could demonstrate the four voice capability, and impressed that he could edit a waveform with the mouse and see it on the scope, but I don't think he was satisfied, since he wanted high quality music. There was a lot of potential in the Mac sound capability, but it would still take years, and the efforts of many third party developers, to fully exploit it.

# The Little Kingdom

**Author:** Andy Hertzfeld
**Date:** December 1982
**Characters:** Dan Kottke, Steve Jobs, Jef Raskin, Mike Moritz
**Topics:** The Press, Personality
**Summary:** Steve Jobs was almost 1982's Man of the Year
**Revision:** most recent of 15




**Steve on the cover of Time**

The February 15th, 1982 edition of Time magazine featured none other than Steve Jobs on its cover, appearing in an article entitled "Striking It Rich: America's Risk Takers". Instead of a photograph, Steve was depicted in a drawing with a red apple balanced on his head that was pierced by a zig-zag bolt of light emanating from an Apple II.

The article inside focused on a number of high tech start-ups, but there was a long sidebar that told the story of Apple's meteoric rise, written by a young business reporter named Mike Moritz. It was a bit critical in places ("As an executive, Jobs has sometimes been petulant and harsh on subordinates"), but in general it was positive about the company and its prospects.

Macintosh development was shrouded in secrecy, even within Apple, so we were surprised one day a few months later when Steve appeared in the software area of Bandley 4 accompanied by the Time reporter, Mike Moritz. Steve requested that I give him a demo of the Macintosh, and answer all of his questions. Apparently, Mike wanted to write a book about Apple, and managed to convince Steve to give him total access to the company, including the Macintosh team.

"Mike's going to be our historian," Steve informed us, "so you can tell him everything. Treat him like he's a member of the team, because he's going to write our story for us."

The previous year, a development team at Data General was immortalized by Tracy Kidder's best selling book, "The Soul of a New Machine", about the ups and downs of developing a new mini-computer. Now it seemed like Mike Moritz was going to do something similar for the Mac team.

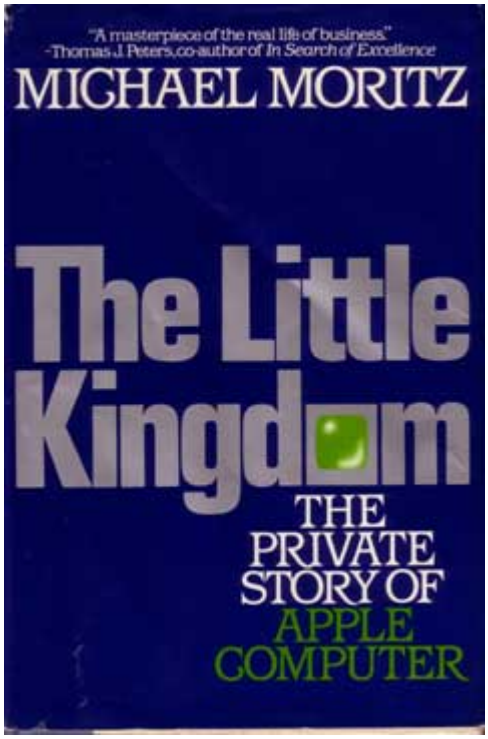
**Mike Moritz in 1984**

Over the next few months, Mike spent lots of time hanging around the Mac team, attending various meetings and conducting interviews over lunch or dinner, to learn our individual stories. Mike had grown up in South Wales and attended Oxford before moving to the US for grad school, obtaining an MBA from Wharton. He was in his mid-twenties, about the same age as most of us, and was very smart, with a sharp, cynical sense of humor, so he fit right in, and seemed to understand what we were trying to accomplish.

In December 1982, word somehow got around that Time Magazine was considering awarding Steve Jobs its prestigious "Man of the Year" designation for 1982. Mike Moritz, who was by now Time's San Francisco Bureau Chief, came down to Apple for another round of interviews, as background for the lengthy "Man of the Year" story. But we were in for a surprise when the award was announced the last week of the year.

Instead of crowning Steve Jobs as the Man of the Year as we expected, Time's editorial staff gave the designation to "The Computer", declaring 1982 to be the "year of the computer" and explaining that "it would have been possible to single out as Man of the Year one of the engineers or entrepreneurs who masterminded this technological revolution, but no one person has clearly dominated those turbulent events. More important, such a selection would obscure the main point. TIME's Man of the Year for 1982, the greatest influence for good or evil, is not a man at all. It is a machine: the computer."

The cover story did include another profile of Steve Jobs, containing some comments that were less than complimentary. One unspecified friend was quoted saying "something is happening to Steve that's sad and not pretty", but the best quote was attributed to Jef Raskin: "He would have made an excellent King of France."




**Mike's book**

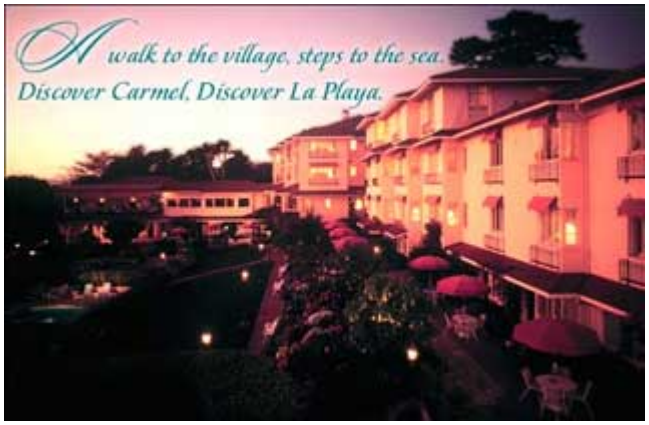
Steve became quite upset when he read an advance copy of the Time article on New Year's eve, and even called up Dan Kottke and Jef Raskin early on New Year's Day to complain to them about it. Soon, Mike Moritz was no longer welcome on the Apple campus; in fact, Steve told the software team "if any one of you ever talk to him again, you'll be fired on the spot!"

But some of us talked with Mike again surreptitiously, as he was putting the finishing touches on his book around the time of the Mac introduction. The book, entitled "The Little Kingdom: The Private Story of Apple Computer", was published in fall of 1984; twenty years later it remains one of the best books about Apple Computer ever written.

Perhaps inspired by the example of Steve Jobs and Apple, Mike Moritz switched careers in 1986 to become a venture capitalist, working for Don Valentine at Sequoia, one of the original investors in Apple. Mike became the original investor in Yahoo in April 1995, convincing Jerry Yang and David Filo to commercialize their web directory, and today is one of the most respected VCs in the industry.

# Credit Where Due

**Author:** Andy Hertzfeld
**Date:** January 1983
**Characters:** Bill Atkinson, Steve Jobs, Bob Belleville, Bruce Daniels, Larry Tesler, Wayne Rosing, Andy Hertzfeld, Rich Page, Rod Holt, Steve Wozniak, John Couch
**Topics:** Marketing, The Launch, The Press, Retreats
**Summary:** Why the Mac design team got credit for their work
**Revision:** most recent of 22



**The La Playa Hotel in Carmel**

The Macintosh team held a series of off-site retreats, every six months or so starting in January 1982. A retreat usually lasted two full days, including an overnight stay. We'd travel by bus to a naturally beautiful resort an hour or two from Apple's offices in Cupertino, like Pajaro Dunes near Monterey Bay. Every employee on the team was invited, as well as folks from other parts of the company who were contributing to the project. The retreats were a mixture of a divisional communications meeting, an inspirational pep talk and a company party, featuring chats with industry legends like Robert Noyce (inventor of the integrated circuit) or Ben Rosen (the VC who funded Compaq and Lotus), and entertainment from Wyndham Hill artists like Liz Story.

The third retreat was scheduled for January 27th and 28th, 1983 at the La Playa Hotel in Carmel, and it came at a pivotal time for the project. The Lisa was just introduced the previous week, after four years of development, on January 19th (although it wouldn't actually ship for another five months), and it was becoming increasingly clear that it was time for the Mac team to shift gears, buckle down and change our focus to doing whatever it took to finish up and ship.

After the two hour bus ride from Cupertino, we gathered in a large meeting room to hear Steve Jobs' opening remarks, which set the agenda for the retreat. Steve was fond of summarizing the themes of the day into a few succinct aphorisms, which he called "Quotations from Chairman Jobs". The sayings from the previous retreat, held in September 1982, were "It's Not Done Until It Ships", "Don't Compromise!" and "The Journey Is The Reward". This time, they were "Real Artists Ship", "It's Better To Be A Pirate Than Join The Navy", and "Mac in a Book by 1986" (see Pirate Flag) .

Even though he was technically a member of the Lisa team, Bill Atkinson attended the Macintosh retreats. Actually, now that the Lisa was finally completed, he planned to shift to working full-time on the Mac, to create a killer graphics application to be bundled with every machine (see MacPaint Evolution) . He was going to start working on it soon, and we were all excited to see what he would come up with.

The first day of the retreat was focused on engineering, and it went by quickly, as each member of the engineering team gave a short talk about their recent and upcoming work in the form of panel discussions, moderated by engineering manager Bob Belleville. At 4PM, the formal part of the meeting ended for the day, and we had a couple of hours of free time to enjoy before dinner.

I was about to join a group going for a walk on the nearby beach when I was pulled aside by Bill Atkinson. It was obvious that something was bothering him.

"Do you have a minute?", Bill asked me urgently, looking kind of somber. "I want to show you something privately." We picked one of the small conference rooms, went inside and closed the door.

Bill was carrying three magazines, which he laid out in front of me on the table. Two of them were very recent issues of personal computer magazines, like Byte and Popular Computing, while the third was more business oriented. They all contained articles about the recently introduced Lisa. He opened one of them and showed me an article, with a sidebar entitled "An Interview with Lisa's Designers".

"Hey, that's cool," I told Bill, "You made it into Byte!"

"Look closer", Bill told me, with a pained expression on his face.

I started to browse the article, and noticed that it interviewed engineering manager Wayne Rosing, software manager Bruce Daniels, and applications group manager Larry Tesler. I finally saw why Bill looked so upset - he wasn't included as one of Lisa's designers, which was absurd, since he did more of the design than everyone else combined.

All three magazine articles featured quotes from Wayne, Bruce and Larry, as well as Steve Jobs and John Couch, the top Lisa executive, but apparently, no one thought to include Bill, even though he designed and implemented the most important parts of Lisa almost singlehandedly, possibly because he wasn't a manager. He was very disheartened, especially because something like this had happened to him once before.

Bill told me that he was haunted by a similar incident that occurred six or seven years earlier. He did some groundbreaking work to create a detailed 3D animation of the human brain. He scanned a series of brain slices, and then wrote software to reconstitute them in an animated sequence, rendering them frame by frame to produce a spectacular movie that depicted important brain structures in stunning detail. The movie won various awards, and a frame from it graced the cover of the October 1978 issue of Scientific American, but one of the professors that he was working for stole most of the credit, acknowledging Bill as only a minor collaborator in the published papers. Now it seemed to be happening all over again.

I tried to cheer him up, telling him that the press was usually wrong about everything anyway, and that everyone at Apple understood his leading role in both the Lisa and Macintosh projects, and that there would be plenty of opportunities to talk with the press in the future. He told me that he was so upset that he was thinking about quitting, unless Apple rectified the situation somehow. We both knew that he needed to talk with Steve Jobs about it, but he was nervous about bringing it up with Steve. I told him that I thought he was completely justified, and that Apple ought to try to make it up to him.

A few hours later, after dinner, Bill told me that he arranged to meet with Steve in private early the next morning, before the day's meetings commenced, but he surprised me by asking me to accompany him. I told him that it wasn't my business, and that I felt that it was inappropriate for me to attend, but Bill insisted, telling me that he needed my support, if only to have someone else present to help ground Steve's infamous reality distortion field (see Reality Distortion Field) . Even though I knew it would be awkward, I told him I would do it.

We both were nervous as Bill knocked on the door of the small office that Steve was using, in the back of the large meeting room where breakfast was being served to the team, at the appointed time. Steve opened the door, but he looked angry when he noticed that I was present.

"What is he doing here?", he asked Bill, before turning to face me. "Go away. This isn't any of your business!"

"No, I need to have Andy here," Bill intervened. "He didn't want to come, but I asked him to be here to support me."

Steve shrugged, and decided to continue as if I wasn't there. "OK, let's hear it, and you need to be quick, because we have to start the meeting soon. What's the big problem?"

Bill explained how upset he was that he didn't get any recognition for his work on Lisa, his voice hesitant at first, but picking up conviction as he started to get emotional. He told Steve that he was thinking about leaving Apple, because he was treated so unfairly.

Even though Steve had enormous respect for Bill, he began to get annoyed, although you could tell that he was trying not to.

"Hey, listen, I'm sorry, but you're overreacting and blowing things out of proportion," Steve replied in a dismissive tone. "Who cares about a couple of magazines? You should have been included, but you weren't. Someone made a mistake. It's not such a big deal."

"That's easy for you to say," Bill retorted, upset at the lack of sympathy. He raised his voice, which was full of emotion. "I'm not going to work here anymore if you don't appreciate what I've done and treat me fairly."

Steve took a step toward the door. He seemed impatient. "I don't have time to deal with this now. We'll straighten it out when we get back. I have 60 other people out there who are pouring their hearts into the Macintosh, and they're waiting for me to start the meeting." He opened the door and left the room without saying another word.

Bill and I remained in the small office, unable to speak, emotionally exhausted from the intense encounter. After a few minutes, we heard a loud cheer, as Steve made a number of announcements to kick off the second day of the retreat. Bill sighed, and we left the office to attend the rest of the meeting.

The following week, Steve arranged for Bill to meet with Apple's HR team, to discuss what was bothering him. Bill reiterated that his main complaint was getting recognition for his work. After more discussions with Steve, they came up with something that was mutually acceptable to everyone.

The solution was to appoint Bill as an Apple Fellow, in recognition for his work on the Lisa. Apple Fellow was the most prestigious technical position at Apple, awarded to only two employees so far: Steve Wozniak and Rod Holt. Now there would be two more, Bill Atkinson and Rich Page, who also made seminal contributions to Lisa. A fringe benefit of being appointed an Apple Fellow was a fresh pile of stock options, which could be quite valuable if Apple's stock price continued to rise.

But most importantly of all, Steve promised Bill that he would receive public recognition for his work on Macintosh. Mac programs had an "About Box", a descriptive dialog box invoked by the first command in the leftmost menu, which would display the author's name. Furthermore, Bill could display his name in the title bar of the main window each time his graphics application was launched. Finally, Steve promised that the Macintosh introduction would acknowledge the folks who actually created the design, rather than the managers who supervised them.

Steve was true to his word, and the seven people that he designated as the "design team" were featured in various ways during the Macintosh launch. Chiat-Day even filmed us for a series of television commercials, which never aired because they were deemed too self-congratulatory. It was fun to get our pictures in the national press (see Can We Keep The Skies Safe?) , but it was also problematic, because there wasn't a fair way to draw the line. At least a dozen individuals made crucial contributions to the design, so there were some hard feelings from the people who didn't make the cut.

In fact, Steve eventually decided that giving recognition to the designers was a bad idea. Nowadays, Apple has abolished programmer names in the "About Box", and closely guards the names of their designers, allowing only a select few employees to interact with the press at all.

# What's A Megaflop?

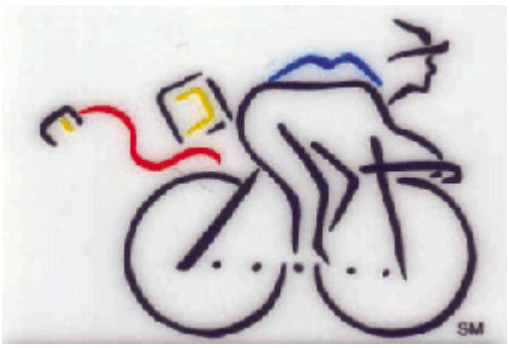
**Author:** Andy Hertzfeld
**Date:** January 1983
**Characters:** Steve Jobs, Joanna Hoffman, Mike Boich, Andy Hertzfeld, Mike Murray, Dan'l Lewin, Andy van Dam
**Topics:** Marketing, Education, Personality
**Summary:** We visit my alma mater to try to sell them Macs
**Revision:** most recent of 14

**Andy van Dam**

Apple always had a natural affinity for education, and, almost from its inception, the Apple II became very successful in the K-12 education market. In the late 1970's, Steve Jobs initiated a marketing program called "Kids Can't Wait" and personally paced the halls of Congress in Washington for three weeks, lobbying for legislation granting tax breaks for donating computers to schools. Even though the national legislation got stymied by politics (it was blocked by Bob Dole), California eventually passed a similar bill and Apple soon donated almost 9,000 computers, one to every school in California.

In early 1982, Joanna Hoffman was still the only marketing person on the Mac team, and she was thinking about which market segments were likely to be early adopters of the Macintosh. She realized that the Mac was almost perfect for college students, and thought it would be worthwhile to put together a plan for selling Macs to higher education.

A few months later, after conferring with a number of consultants who understood the college market, a plan began to emerge. One of the words that the consultants reiterated was "consortium"; it seemed like colleges loved to band together into various consortiums. We knew that the paucity of software at launch would be a barrier to initial acceptance, but maybe not if we could get the colleges to form a Macintosh consortium, where members received steeply discounted Macs for students and faculty. All we had to do is sign up a few of the most prestigious schools, and many of the rest would follow.



**University Consortium Logo**

Mike Murray (who was now the permanent interim Macintosh marketing manager) and Joanna realized that they needed a superb salesperson to take charge of recruiting customers for our consortium-to-be. The best salesperson that Joanna knew on the Lisa team was Dan'l Lewin, a handsome, personable, ivy educated ex-competitive swimmer who was frustrated with his current job of selling the Lisa to corporations. Dan'l was intrigued, and, after some negotiation, was soon barnstorming around the country visiting the leading universities, with Mike Boich in tow to run the demo and answer technical questions, trying to convince universities to sign up with Apple and buy discounted Macs by the thousands.

Some of the universities, like Drexel University in Philadelphia, were easy sells, since they were already thinking about buying a computer for each freshman, and the Macintosh consortium was the answer to their prayers. But others weren't that enthusiastic, and required lots of hand-holding to coax them into the fold. But slowly Dan'l was able to build up a fairly impressive roster.

Toward the end of January 1983, I was asked to accompany Dan'l and Mike on one of the more unusual sales calls, to Brown University in Providence, Rhode Island, because Brown was my alma mater. I hadn't been back there since I graduated in 1975. Brown had a strong computer science program, especially in computer graphics, and was considered to be rather influential with the other universities. They had recently splurged, buying dozens of powerful Apollo workstations, costing tens of thousands of dollars apiece, so we were afraid they'd think the Mac was underpowered. It was thought to be so important that even Steve Jobs agreed to come along as part of the Apple contingent.

The most influential decision maker on the Brown faculty was a computer science professor named Andy van Dam. I was one his teaching assistants during my senior year, so I got to know him pretty well. He was high strung and hard driving, and a little bit like Steve in his tendency to think that the universe revolved around him. I thought that it would be interesting to see how they interacted.

Introductions were exchanged, and we were taken for a tour of the Brown Computing Lab, where they proudly showed off their brand new Apollo workstations. Then we were ushered into a conference room, where Dan'l talked about the consortium while we set up the Mac. We put it through its paces for the benefit of a half dozen faculty members and grad students, with Steve doing most of the talking, eliciting oohs and ahhs in all the right places. Finally, the demo was over and Steve asked them what they thought.

Andy van Dam cleared his throat and looked right at Steve. "Well, its really impressive, Steve, and of course we'll want to join your program. But it's not exactly what we've been waiting for."

Steve looked a little angry. "What are you waiting for? You're going to have to wait a long time to find something better than the Mac!"

"Well, 128K isn't nearly enough memory to do what we want, not even close, and the screen is just too small. We're waiting for a 3M machine, and most of the other colleges are, too."

"A what?"

"A 3M machine. There was a recently published paper that coined the term. You know, a workstation with at least a megabyte of memory, a million pixel display, and a megaflop of computational horsepower. We believe that's what we need for an effective educational workstation."

A megaflop was the ability to execute one million floating point operations per second. The Mac didn't have any floating point hardware, so it was off by an order of magnitude. In fact, we were off by around a factor of 10 in each of the three dimensions.

"Oh, we believe in that, too," Steve shot back, without skipping a beat. "Apple will have an affordable 3M machine before anyone else. I only have one question. What's a megaflop?"

# Little Rubber Feet

**Author:** Paul Tavenier

**Date:** January 1983

**Characters:** Jerry Manock, Paul Tavenier, Steve Jobs, Matt Carter

**Topics:** Industrial Design

**Summary:** When industrial design and the reality of purchasing meet, and we find that no detail was too small to cause problems...

**Revision:** most recent of 7

I was recruited to do the mechanical purchasing on the Mac in Dec of 1982. I was also one of a very few of the "Bozos" that were allowed to transfer over to the project from the manufacturing side of the Apple II business, as Steve Jobs had an extremely low regard for most everyone who worked there, and almost always hired from outside.

At a shortage meeting it was mentioned that the "little rubber feet" that mounted on the bottom of the case were not sticking where they belonged, and there was also trouble with availability from the supplier, Trend Plastics. In investigating this, I found that there was tooling that molded a custom rubber foot with a recessed Apple logo in it. Trend was a great supplier, but this part was a little unusual for them and they were having trouble sourcing the proper peel-and-stick adhesive for the job. We had paid about 8k for the tool, and the parts were something like two bits each.

At the next day's meeting, I mentioned that perhaps the design was a little overboard and maybe we should reconsider, as 3M had a standard part, called a "bump-on", that was available in the correct size, stuck to the case properly, and could even be molded in Apple beige if we desired. Cost was less than 2 cents.

OK, maybe what I really said was "This part is a real designer's wet dream, we need to lose it."

Jerry Manock, who was in charge of the industrial design group, was not even remotely amused. He wasn't even at the meeting, but someone ratted me out. (I would have made the comment even if he was there, that's just the way things were - you really could speak your mind.) He had also been contacted by the 3M rep, as that was the first guy I had called when chasing down shortages.

A very angry Jerry dragged me into a conference room. "Who authorized you to contact 3M about this part", he demanded. "I don't appreciate the wet dream comment either".

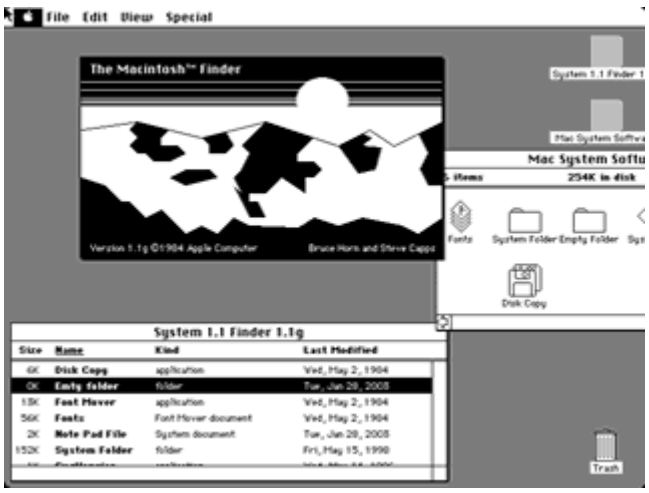
I replied that my job was to fill the shortages for the pilot build, and that I didn't need authorization from him or anyone else to contact a supplier. If he had a problem he could take it up with Matt Carter, who was then in charge of manufacturing.

We then called a truce, and went on to pretty much tolerate each other after that.

Trend made one more attempt to get the adhesive right and failed. Cost considerations won out, and every Mac shipped from Fremont left the plant with four 3M bump-ons stuck firmly to the bottom.

# The Grand Unified Model (2) - The Finder

**Author:** Bruce Horn
**Date:** January 1983
**Characters:** Bruce Horn, Larry Kenyon, Steve Jobs, Frank Ludolph, Steve Capps, Dan Smith, Bill Atkinson, Andy Hertzfeld
**Topics:** Software Design
**Summary:** The evolution of the Finder
**Revision:** most recent of 7




**The Finder**

One of the first things I did when I joined the Mac group was to begin working on the Finder. The first Finder, written in early 1982 with Andy's help, was a simple diskette image with tabs that represented the files on the disk. (see Early Demos) . This Finder was the first to begin to take advantage of the idea of spatial organization: you could drag the tabs around and place them wherever you wanted on the floppy image. Also, my experience with Smalltalk showed through: the big "Do It" button was named after the Do It menu item in Smalltalk, which evaluated a selected expression. This Finder was actually usable, and served as a placeholder until the real Finder was available.

Immediately after the first Finder prototype, I wrote a second which was much more recognizable as the ancestor to the Finder that shipped. This prototype was a nonfunctional prototype that did not actually read the disk, but instead, read a text file that described a hierarchy of files within folders that would be displayed in windows. Our filesystem at the time did not have the concept of directories, so I had to fake it with the proof-of-concept prototype. This was the first Finder that provided double-clicking to open folders, documents, and applications; drag-and-drop to move files between folders; icon and list views; and persistent spatial locations of icons within windows. Of course, it was all window-dressing and none of it was functional, but it did give a good idea of what we would eventually want to implement. Unfortunately, it also made it look like the actual Finder implementation would be easy, which it most definitely was not.

Bill Atkinson came by and I gave him a demo. He had been thinking about the Lisa Filer, which was being written by Dan Smith and Frank Ludolph, and was dissatisfied with its design. When he saw in our Finder mockup some of the ideas that he had also seen in a MIT project called Dataland, he was convinced, and the IF (Icon Filer) project was born (see Rosing's Rascals) . Bill, Dan and Frank put together a new Filer based on these concepts in time to ship with the first Lisa in 1983. In the meantime, I was working on the Resource Manager until later that year.

But I still couldn't get started on the Finder until I figured out how to handle files and applications. We were trying to make the Macintosh a very friendly computer, an information appliance, something that everyone could use. For example, one of the things that I felt could stand improvement in the current computing experience was the problem of filenames. In the Finder, I wanted to make it as easy as possible to give meaningful names to files without excessive restrictions placed on them.

At the time (and still, in some cases, now) filenames were very restricted, both in length and in format. Filenames had to have a three character suffix, with a dot, to denote their file types: text files were named "myfile.txt" and executable applications were named "word.exe". Filenames were also typically limited to eight characters, not including the suffix; this led to very cryptic naming on other computers, which we definitely wanted to avoid.

We decided that we needed to allow users to name their files whatever they wanted, with any characters, including spaces. Because the Finder would allow the user to simply click on a particular file to choose it, special characters like spaces would be no problem; in command-line systems, parsing filenames with special characters could be problematic.

The Grand Unified Model provided a framework for solving this problem too. Since resource objects were typed, indicating their internal data format, and had ID's or names, it seemed that files should be able to be typed in the same way. There should be no difference between the formats of an independent TEXT file, stored as a standalone file, and a TEXT resource, stored with other objects in a resource file. So I decided we should give files the same four-byte type as resources, known as the type code. Of course, the user should not have to know anything about the file's type; that was the computer's job. So Larry Kenyon made space in the directory entry for each file for the type code, and the Mac would maintain the name as a completely independent piece of information.

Simply storing the file type in the directory was not enough, however. There might be many different applications that could open files of a given type (say, a text file); how would the Mac know that a text file called "My Resume" needed to be opened in MacWrite, and another text file called "Marketing Plan" needed to be opened in WriteNow? Just knowing the file's type wasn't enough; the Finder also had to know which program created the file, and thus would be the best choice to open it. Thus another four-byte "creator code" would also be maintained, which would tell the Finder which program needed to be launched to open a particular document. For convenience, the user could also easily override this default, by dragging the document to whichever application he would like.

Finally, we also wanted to have useful and meaningful icons for programs and documents on the Mac. Using the type and creator mechanism, this was easy; we would just associate a specific icon for each file type that is handled by a particular application. Given a (type, creator) pair, it would be easy to look up the appropriate icon to draw for the file.

But where would these icons come from, and where would they be stored? It seemed clear that each program would be responsible for defining icons for the application and its documents, and that this information should be stored in the application itself; but if we simply opened the application's file each time we needed to draw an icon, the Finder would be terribly slow. I decided that the Finder needed to cache these icons and associations in a resource file. This was the Desktop Database.

**The Desktop Database**

Programmers reading Macintosh technotes on the Resource Manager would run across the statement that "The Resource Manager Is Not A Database!" This is ironic, because one of the very first uses of the Resource Manager was in fact for a database: the Desktop Database, which stored and maintained information relating to applications, documents, and their respective icons.

Applications would include a set of resources with known ID's in their resource map; this group of resources would collectively be called the application's Bundle, and it would have the type BNDL. In the bundle were special FREF resources (file references) that would map a document or application type to an icon ID.

When the Finder would encounter an application on a disk, it would look into the Desktop Database to detemine whether it had already seen this application and copied its bundle. If not, it would copy the bundle and associated icon and FREF resources, and reassign the ID's of the icons so that they would be unique within the database. The FREF resource provided an indirect ID mapping to make this possible. In this way, all bundles could be readdressed within the database and there would be no conflict between the different resources pulled in from a variety of applications.

Drawing the image of a particular document was then a simple matter of looking up the appropriate icon by type and creator in the Desktop Database, and loading the resource. On the 400K disk, this worked very well--the Resource Manager was up to the task, and it wouldn't show its limitations for several years--an eternity in the software world.

**The Final Push**

By late 1982, the Lisa Filer was nearly finished. At 360K, it was a significant application--much too large for us to use, and it wasn't based on the Grand Unified Model of types, creators, and resources. Although Bill offered us the code to the Filer, we always knew that we'd be writing our Finder from scratch.

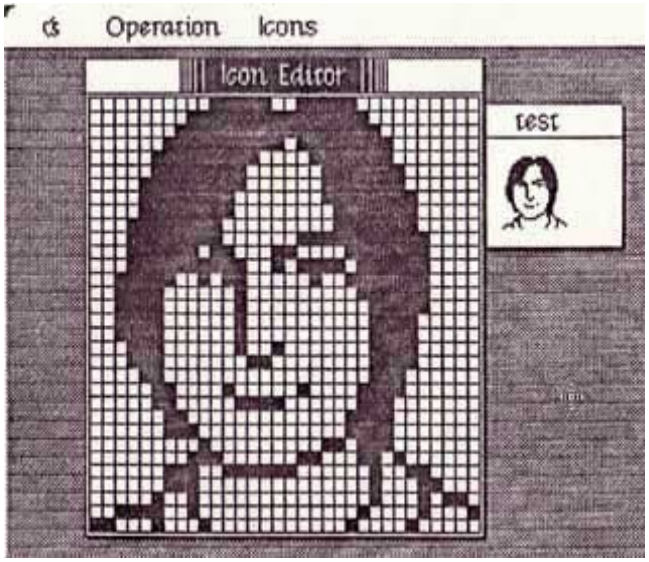
The Mac was to ship in January, 1984, our final drop dead date. In late 1983 it became clear that I needed help to finish it in time, and Steve Capps was recruited to work with me on the Finder (see Steve Capps Day) . Also by this time, reporters, writers, and associated VIPs were being escorted through the Mac Group to see our new machine, and to talk with Andy, Steve Jobs, and some of the other folks on the team. Capps and I had too much work to do to be distracted by the reporters, so we moved to a separate office in a different building.

Writing the Finder was not easy. Because of the tight memory requirements, most of the code was 68K assembly, like the ROM Toolbox. Capps took on some of the more difficult tasks, including file copying (see Disk Swapper's Elbow) . We worked late into the night, listening to Violent Femmes and Capps' other punk rock recordings. The clock was ticking.

Capps and I ended up finishing version 1.0 of the Finder in time for the Mac to be introduced on January 24 (see Real Artists Ship) . It weighed in at 46K bytes, approximately 1/8th the size of the Lisa Filer. The small size of the Finder made it possible to have the entire System and Finder, plus an application, plus a few documents, on a single 400K floppy...just barely. We were overjoyed, but exhausted. The Grand Unified Model --resources, types and creators, the Desktop Database, and the Finder-- was finally done

# Steve Icon

**Author:** Andy Hertzfeld
**Date:** February 1983
**Characters:** Susan Kare, Steve Jobs, Bill Atkinson, Burrell Smith
**Topics:** MacPaint, Personality
**Summary:** Having your own icon became a status symbol
**Revision:** most recent of 14




**The Steve Jobs icon, by Susan Kare, in the Icon Editor**

In February 1983, I worked on putting together an icon editor for Susan Kare to use to create icons for the Finder. Inspired by the "Fat Bits" pixel editing mode that Bill Atkinson had recently added to MacPaint, it had a large window with a 32 by 32 grid, displaying each pixel at eight times its natural size, as well as a small window that showed the icon at its actual size. Clicking on a pixel would invert it, and subsequent dragging would propagate the change to the dragged over pixels.

Susan started working on icons for the Finder, but she also would draw lots of other images as well, for practice or just for fun, usually reflecting her whimsical sense of humor. One day, I came over to her cubicle to see what she was working on, and I was surprised to see her laboring over a tiny icon portrait of Steve Jobs.

Icons were only 32 by 32 black or white pixels, 1024 dots in total, and I didn't think it was possible to do a very good portrait in that tiny a space, but somehow Susan had succeeded in crafting an instantly recognizable likeness with a mischevious grin that captured a lot of Steve's personality. Everyone she showed it to liked it, even Steve himself.

**Bill icon**

Bill Atkinson was so impressed with the Steve icon that he asked Susan to do one of him, that he could use in the MacPaint about box. He sat in Susan's cubicle for an hour or so, chatting with her while she crafted his icon. I don't think it turned out quite as good as the Steve icon, but it certainly was an unmistakable likeness, and did become part of MacPaint.

At that point, It became a Mac team status symbol to be iconified by Susan. As soon as he saw Bill's icon, Burrell Smith started begging Susan for a Burrell icon, even though he had no specific use for it. He lobbied Susan for a few days, making his standard offer of best friendship (see I'll Be Your Best Friend) , before she gave in and had him pose for his icon. Unfortunately, I can't find a copy of the Burrell icon to display here.

Susan did a few more portraits, for various members of the team who desired to be immortalized in a thousand dots. She'd usually work on them in the late afternoon, chatting with the subjects as they posed, while other team members listened in. I got to know a few of my teammates a lot better from these sessions.

# Too Big For My Britches

**Author:** Andy Hertzfeld
**Date:** February 1983
**Characters:** Bob Belleville, Steve Jobs, Jerome Coonen, Andy Hertzfeld, Steve Capps, Larry Kenyon, Pat Sharp, Rod Holt
**Topics:** Management, Personality Clashes
**Summary:** My belated performance review is delivered verbally
**Revision:** most recent of 19

Apple's HR policy dictated that each employee was supposed to receive a performance review from their manager every six months, which helped to determine your salary increase or possibly an award of additional stock options. But as the end of 1982 approached, I hadn't received my review for more than eight months.

This wasn't very surprising, since Bob Belleville, who was my boss and our software manager, was not getting along very well with the software team. He thought that some of us were intrinsically unmanageable, and that we didn't sufficiently respect him. Bob had replaced Rod Holt as the overall engineering manager in August, responsible for both hardware and software, and had just hired a new software manager, Jerome Coonen, who was slated to begin in January, which would allow him to further distance himself from the software team. But he still had to deal with us directly one last time to write our reviews for 1982.

By the end of January, everyone on the team had received their review except for me. Others mentioned that Bob had acted somewhat strangely during their reviews, making cryptic remarks that they didn't understand, so I wasn't particularly looking forward to mine. I occasionally had to interact with Bob, but he was reticent around me, not saying much, seemingly hiding behind his enigmatic, tight-lipped smile. Finally, after another couple of weeks, Bob's secretary called me to arrange an appointment, presumably for my belated review.

The meeting was scheduled for 5pm, toward the end of the day on a Thursday afternoon in mid-February. Bob was waiting for me when I entered his corner cubicle. I asked him what was up. He said that he didn't want to get into it in the office, and suggested that we discuss it on a walk around the block. That was fine with me, but now I was even more apprehensive - generally, walks around the block were reserved for firing or demoting someone, or to talk someone into staying after they had quit.

Bob waited until we were a full block away from Bandley 4 before starting to speak.

"Well, Andy, you're not going to like hearing this, but you're a big problem on the software team and I'm giving you a negative review for the last six months of 1982."

I knew that Bob disliked me, but I was nevertheless shocked. I was working my heart out for the last two years, devoting my life to the Macintosh, seven days per week, holding the project together after Bud returned to medical school. I was really doing two full-time jobs, writing the Mac Toolbox in assembly language by night and helping everybody else by doing whatever was necessary each day.

"How can you say that?", I responded, horrified. "I accomplished everything that I was supposed to, and a lot more besides". All my previous reviews from Apple were extremely positive, including the last one from Bob, so this was new to me.

Bob unfurled his mirthless grin. "Oh, don't get me wrong, I think your technical work was perfectly adequate during the review period, and I don't have a single criticism of it. That's not your problem area. I don't have a single complaint with your technical work." He paused for a moment, to take a deep breath, and then continued.

"The problem is with your attitude, and your relationship with management. You are consistently insubordinate, and you don't have any respect for lines of authority. I think you are undermining everybody else on the software team. You are too big for your britches."

At this point, as he probably expected, I broke down into tears. The Macintosh was at the center of my life, and it was suddenly clear that I was going to have to quit. I couldn't work for somebody who was saying this, no matter how much the project mattered to me.

Perhaps Bob was a little taken aback by my tears, so he tried to soften things. "Listen, this could be a very expensive conversation. It could turn out to be either very good or very bad for both of us. I'm trying to get you to see how if you listen to me, things could turn out very good for both of us."

I had no idea what he was talking about, or how a bad review could possibly be good for me. "What do you mean, undermining the team?", I managed to choke out, "I'm always trying to help everybody else on the team. Give me one example of someone that I've undermined."

"Larry Kenyon", Bob replied. "You're stifling Larry Kenyon. Now he is someone with a good attitude, and you're keeping him from realizing his potential."

I always thought that I had gotten along great with Larry. I recruited him to the Mac team, after working with him on Apple II peripheral cards in 1980, and then handed off the low-level OS stuff to him while I worked on the Toolbox. I thought Larry was a terrific programmer and a great all around person, and treated him with the highest respect, and always enjoyed it when we worked together. I think I knew what Bob was getting at, though, as I had reacted poorly a few months ago when Bob appointed Larry as temporary manager when he had to be gone on a short trip, probably just to irk me.

By this point, I was crying harder, and Bob looked like he might start crying at any moment now, too. We were also pretty far from Bandley 4 by now, and it was starting to get dark. The tone of the conversation seemed to shift as we both realized that we should start heading back.

"This doesn't have to be that bad", Bob said as we turned around. "All you have to do is listen to me and things will work out fine."

"What do you mean?", I asked him.

"You need to show more respect to authority. It's not just me. Jerome is still new, and I'm afraid that you won't let him do his job. He's your boss now, and you need to show him respect, and let him do his job . But that's not the main problem. What you really have to do is stop talking to Steve Jobs." Bob paused and flinched slightly, as if just mentioning Steve was difficult for him.

"Whenever there's something that you don't like, even little things, you go running straight to Steve, and he interferes. I don't have any authority with the software team, because they always hear everything from Steve before I do, and he always hears everything about the software straight from you. It's making it so I can't do my job. You should communicate through the proper channels. I can't tell Steve what to do, but you work for me, so I can tell you."

I did respect Jerome, and I was trying to make an extra effort to support him as our manager, because I knew that we really needed him. Jerome was a very smart guy, and a passionate genius when it came to numerical software - I loved to hear him elucidate the intricacies of his beloved floating point routines. But I did consider him to be more of a partner than a boss, just like I did everyone else on the team, but I didn't think that he had a problem with that. But apparently Bob did.

But the Steve issue was different. From the earliest days of the project, Steve would usually show up at the Mac building in the late afternoon, or sometimes after dinner, and ask us about the happenings of the day. We would demo the latest stuff to him, or he'd complain about something, or sometimes we'd just exchange the latest gossip. After Bud went back to medical school, Burrell and I were the only ones who would regularly stay late, but after a while, more of the team began to hang out with us. It wasn't unusual for six or eight of us to go out for a late dinner, and then come back and keep working. By early 1983, most of the software team was staying late, and even some marketing and finance people would join us, but Bob Belleville never did, since he had to get home to his wife and two young daughters.

"I can't stop Steve from coming around", I told Bob. "If you don't want me to talk with Steve, you're going to have to tell him about it. I like Jerome, and I have no problem working with him, but now it looks like I have a problem working with you. If you think that I'm undermining the team, I'm out of here tomorrow."

Bob looked at me intently. "I don't have the power to fire you", he said. "You're going to give me power that I don't have if you quit. Do you really want to do that?"

By now, it was completely dark as we were approaching the Apple parking lot. We stopped in front of Bob's car.

"This could be a really expensive conversation for both of us", Bob muttered cryptically. "It's entirely up to you." With that, he got into his car and drove off, and I wandered back into Bandley 4, feeling stunned and drained. I got back to my cubicle, put my head down on the desk, and started crying again.

It was around 6:30pm now, and most of the software team was still around. Capps saw that I was upset, and asked me what was wrong. He began to get angry when I told him and a few others what happened, and he made me promise not to overreact until he had a chance to find out what was going on.

Larry Kenyon was still in his cubicle, so I went over and told him what Bob had said. I asked him to be honest, and to tell me if he thought I was stifling him in any way.

"You've got to be kidding!", Larry exclaimed. "I think it's really great working with you, that's the reason I'm on this team. I think it's an honor to work with you." With that, I burst into tears again, touched by Larry's support.

I was exhausted and confused, so I went home to get some sleep and to think about what I should do next. When I came in earlier than usual the next morning, there was a message on my desk to call Pat Sharp, Steve's secretary. She told me to come by his office right away because he wanted to talk with me as soon as possible.

"I can't believe Bob gave you that review", Steve started talking even before I stepped into his office. "He showed it to me a week ago, but I refused to approve it, and I told him to write something more positive. Do you have a copy of it?"

I told Steve that he didn't give me a copy, he just delivered it verbally. I told him that I couldn't work for someone who feels that way about me, and that I had no alternative but to quit.

"It's good that you don't have a copy, because that review is rescinded, it doesn't officially exist. I just got done talking with Bob, and after I chewed him out, he also quit, because he said that he can't manage the software team. And Capps came in here and told me that the rest of the software team is so upset that they're thinking about quitting, too. What a mess! You and Bob don't have to love each other to work together. We're going to sit down this afternoon and talk this thing out until it's resolved."

So, at 4pm on Friday afternoon, as soon as Steve was available, the entire software team, plus Burrell, filed into one of the conference rooms. We all sat in a semi-circle of chairs on the right side of the room, waiting apprehensively. Finally, Steve strode in, with his characteristic bouncy stride, trailed by a despondent looking Bob Belleville, who took a seat on the left side of the room, facing the software team.

Steve started talking first, saying that tensions had been building up for a while, and it was time to clear the air, so we could all pull together down the home stretch. All the while, Bob was staring at the floor, unwilling to make eye contact with anyone else, controlling his emotions behind a tight-lipped expression, halfway between grin and grimace.

"OK, who's going to go first? What's the problem, and how do we fix it?", Steve asked.

Capps spoke up, explaining how painful and unmotivating it was to see me broken up about an obviously unjustified review. He wanted to know how things could have gotten so screwed up.

Steve nodded to Bob, encouraging him to speak up. Bob spoke in a monotone. "I didn't give Andy a bad review. I told him that his work was fine."

I was flabbergasted. "You said I was undermining the team, and stifling Larry," I blurted out. "I can't work for you if you think I'm undermining the team."

Bob looked up, looking me in the eye for the first time. He spoke in a mild, low emotionless monotone. "I didn't say any of those things. Why are you claiming that I said that?"

I was shocked. Bob was denying what he told me the day before, and I know that I didn't imagine it. Furthermore, he really seemed to genuinely believe what he was saying, and he looked to be in a kind of trance, both depressed and confused. If he didn't acknowledge what he said to me, there was no way to resolve it. I didn't know how to proceed, so I backed off my accusations.

A few more people spoke up, addressing other grievances, but Bob's trance-like manner persisted and eventually the meeting broke up, without anyone being satisfied. Steve tried to declare victory at the end, but no one was buying it.

I thought about things over the weekend, and realized that I cared too much about the Macintosh to quit before it was finished, managerial adversity notwithstanding. The situation that I feared when Bud left had actually occured, in spades, and I wasn't confident that Steve would live up to his promise to protect me. I wasn't sure what was going to happen, but I knew that my blissful days at Apple were over, and that things were going to be different from now on.

# RMaker

**Author:** Andy Hertzfeld
**Date:** March 1983
**Characters:** Andy Hertzfeld, Bob Belleville, Bruce Horn, Dan Cochran, Jerome Coonen, Bill Duvall, Barry Haynes
**Topics:** Technical, Software Design, Prototypes, LaserWriter, Lisa, Lisa Rivalry
**Summary:** A wild hack is added to the resource compiler

The resource manager became stable enough for other parts of the Toolbox to use in the fall of 1982. At first, only the dialog manager (which was also written by Bruce Horn) used resources, but soon they began to spread inexorably throughout the system. By early 1983, we were using resources to define windows, menus and controls, but we didn't have a decent way for developers to specify them.

Bruce had written a "dialog compiler" that read in a terse text-based description of a dialog and created the corresponding dialog resources. In February 1983, I expanded it into a general resource building tool that I called "RMaker", which ran on a Lisa and built a resource file from a text-based description. RMaker allowed arbitrary resources of any type to be specified in hexadecimal, and provided some specialized formats for specific types like windows or menus. For an example of an RMaker source file, click here to see MacPaint's resource definition file.

The Macintosh was built around the Motorola 68000 microprocessor, which was an amazing chip for its day, but it did have a few problematic limitations. The instruction set was missing a way to specify a long relative branch, and absolute branches were forbidden since we needed our code to be position independent. This meant that the maximum size of a hunk of code was limited to the range of a relative branch, which was 32K bytes.

That wasn't nearly enough big enough for Lisa applications, so the Lisa development system supported multiple code "segments", stitched together with a jump table for inter-segment references. Since the Macintosh had much less memory than the Lisa, I thought that we could forgo more complicated machinery and get away with supporting only a single code segment per application, and that's what we did at first. But of course I was wrong.

In March 1983, Jerome Coonen came to me with a worried look on his face. Apparently, three different developers started running up against the 32K code size limitation all at once, and he didn't know what we could do about it. The schedule was already barely achievable, and it didn't seem possible to implement a multi-segment loader in the required time frame. He asked me what I thought we should do.

Even though I hadn't thought much about it before, it seemed clear to me that we should keep the code segments in resources, since that's what we were doing with just about everything else these days. We'd have to write a little bit of tricky code to create, load and maintain the jump table, but that didn't seem so bad. I told Jerome that I thought I could have a resource-based multi-segment loader going in less than a week.

Jerome was surprised, thinking that it would take at least a month and asked me to get going on it right away. I was a little hesitant, since my bad review from Bob Belleville a week or two ago was still fresh in my mind (see Too Big For My Britches) , which made me less inclined to put in another heroic effort, but as usual I got excited about the new approach and dove in to try to accomplish it.

The hardest part of the job was writing a tool to extract the code from the output of the Lisa linker and turn it into code resources. I was still working on polishing RMaker, and I realized that it was the perfect place to do the job. I added some code to RMaker to build code resources and the jump table from object files, and was able to get a proof of concept multi-segment loader working in just a few days.

The Lisa linker supported two different sizes of jump table entries. There was a 6 byte version, that just consisted of a single long jump instruction, to an absolute but virtual memory address, which relied on the memory management of the underlying OS to load the appropriate code. The Macintosh didn't have any memory management hardware, so that wasn't possible for us. Luckily, the linker also supported a 10 byte version, which didn't rely on virtual memory and made an explicit OS call to load the segment, which is what I used for my initial version.

It took four bytes to invoke a Lisa system call, but the Macintosh could do it in 2 bytes using a 'trap' instruction, so we really only needed 8 bytes per jump table entry. 8 bytes was a lot better than 10, since not only did it save precious memory, but it would keep everything aligned to 32-bit boundaries, which would eventually be important. I thought it would be simple to modify the Lisa linker to support an 8 byte/entry jump table variant, but I didn't have the source code, so I requested that the Lisa group make the change for us.

But the engineer on the Lisa team that was responsible for the linker, who will rename nameless in this account, wasn't a fan of the Mac project and claimed that he had other, higher priority tasks to accomplish before he could get around to it. He also refused to give me the source code, so I couldn't simply do it myself. After a week or so, I grew tired of waiting, and thought of a wild hack that would let me move forward without him.

The problem was that the linker generated jump table offsets for every inter-segment procedure invocation. Since it thought that the jump table had 10 bytes/entry, I had to find and correct every jump table reference, multiplying it by 8/10 to adjust it for the 8 byte/entry jump table. The jump table was only referenced by a few, specific op-codes, so I added a routine to RMaker to scan through the object code, identifying jump table references and patching them. This wasn't bullet-proof, since the patch routine couldn't tell code from in-line data, so it might inadvertently patch something it shouldn't, despite the extra sanity checks that I threw in. But I thought I could get away with it for the short while before a modified linker became available.

Even though I made repeated requests, the Lisa linker engineer didn't implement the 8 byte version of the jump table for more than nine months. In fact, he never did, but in early 1984, he transferred to another project, and Barry Haynes, the engineer who replaced him, was able to accomplish it a few days after taking over the job, just as I was going on my leave of absence in March 1984, so we were finally able to eliminate RMaker's ugly hack. It was a miracle that no one ever stumbled over it, at least as far as I know.

I went on leave of absence from Apple in March 1984 (see Leave Of Absence) , but I still was excited about writing software for Macintosh as an independent developer, and was full of ideas about different programs to write. But I wasn't happy about the fact that you still needed a Lisa to write software for the Mac; among other things, the recent linker problem had left a bad taste in my mouth regarding the Lisa development system. I had a Lisa at my house, but it really belonged to Apple, and I didn't want to have to buy one. I decided to try to cobble together an entirely Mac-based development system.

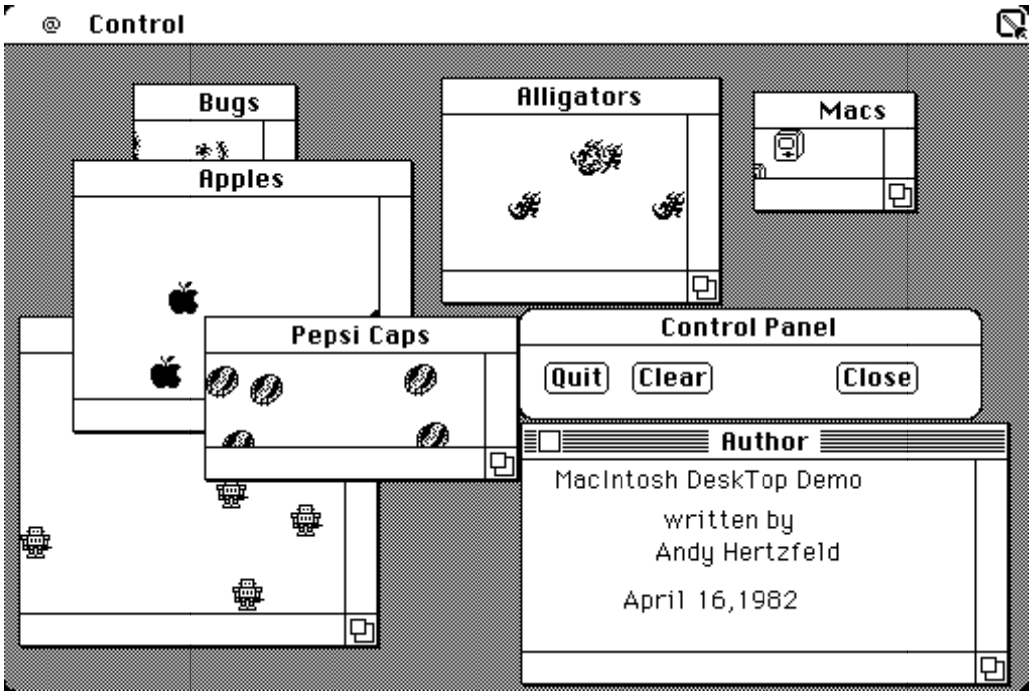
Bill Duvall was an old friend of Bob Belleville's from Xerox who we seeded with a Mac prototype in the spring of 1983 so he could port his development system to it, including an assembler and a C compiler, to be sold by his own tiny company named Consulair. His linker even generated Macintosh code resources, but there was no equivalent of RMaker for building the other resources

required by a Mac application - you still had to do that on a Lisa. I decided to rewrite RMaker for the Mac, so I could finally abandon the Lisa. I used assembly language, since the assembler was much further along than Consulair C was at the time. In a week or so, I had a new RMaker that was integrated with the Consulair tools, so you finally could write a complete application with just a Macintosh.

In the summer of 1984, I got a call from Dan Cochran, who was the marketing person at Apple in charge of developer tools. He had heard about my new version of RMaker, and wanted to bundle it with the Consulair package. After a bit of negotiation, I sold it to Apple in exchange for a brand new LaserWriter prototype. I drove up to Bill Duvall's house in the Berkeley hills to help him incorporate RMaker into his system. When my reward arrived in October, I thought it was pretty cool to be the first on my block to have a LaserWriter, three months before it was officially announced.

# Bouncing Pepsis

**Author:** Andy Hertzfeld
**Date:** March 1983
**Characters:** Andy Hertzfeld, Steve Jobs, John Sculley, Susan Kare, Bill Atkinson, Mike Murray
**Topics:** Software Design, Recruiting, Early Programs
**Summary:** We cook up a special demo for John Sculley
**Revision:** most recent of 3




**We added Pepsi Icons for John Sculley's visit**

The Window manager was the one of the most important parts of the User Interface toolbox, and the ultimate showcase for Quickdraw's "region clipping" technology. The window manager had to calculate various regions for windows as they were created, moved and resized, and objects drawn inside the windows were automatically clipped as necessary.

The Macintosh window manager was based on the one that Bill Atkinson wrote for Lisa, which was written in Pascal; my job was to rewrite it in 68000 assembly language and adapt it to the Macintosh environment. The first step was to port Bill's Pascal version. I wrote a little program to test the port, which I called "Window Manager Demo", that generated some windows and put the window manager through its paces.

A year earlier, I had written a fast "ball bouncing" routine, using custom, 16x16 pixel graphics routines, that could animate hundreds of balls simultaneously, which was a fun way to show off the Mac's raw graphical horsepower. I decided to animate a few dozen balls in each window in the window manager demo, using Quickdraw, because their continuous movement would eventually cover all the bases inside a window and expose any flaws in the underlying clipping.

After Susan started in January 1983, I asked her to draw some tiny 16 by 16 bitmaps to use in the Window Manager Demo instead of the by now monotonous ball shapes. Soon, we had a variety of little objects bouncing around in the various windows, like tiny little Macintoshes, or apples, insects and alligators. I thought that the Window Manager Demo was finished, but I was wrong.

Steve Jobs came by the software area one evening a couple of months later, excited about someone he had recently met in New York City. "Hey, I want you to do a demo next week for this guy I met yesterday, John Sculley, he's the president of Pepsi. He's really smart - you wouldn't believe how smart he is. If we impress him, we can get Pepsi to buy thousands of Macs. Maybe even five thousand. Why don't you try to come up with something special to show him?".

It sounded a little bit fishy to me, since we hardly ever demoed to potential customers at that point. But I asked Susan to draw some Pepsi imagery, and she came up with tiny little Pepsi caps, as well as Pespi cans, in his honor, so I put them into the Window Manager demo.

The next week, Mike Murray led John Sculley around the engineering area, since Steve was out of town. He brought him by my cubicle to see the modified Window Manager demo. I opened the windows one at a time, saving the Pepsi caps and cans for last. He seemed genuinely excited to see the Pepsi stuff, but oddly cold for most of the demo. He asked a few questions, but he didn't seem all that interested in the answers.

A few weeks later, we found out the real story - the purpose of John's visit was to interview for CEO of Apple, and he got the job, being convinced by Steve's famous line "Would you rather sell sugar water to kids for the rest of your life, or would you like a chance to change the world?".

# A Big Scare!FIXXXXXXEGEE foto

**Author:** Dan Cochran
**Date:** April 1983
**Characters:** Jerry Brown, Dan Cochran, Steve Jobs, John Sculley, Al Mandel
**Topics:** Celebrities, Lisa
**Summary:** A visit to the Lisa offices by former Governor Jerry Brown
**Revision:** most recent of 7

n April 8th of 1983, Steve Jobs came by the Lisa offices on Bandley Drive with the former governor of California, Jerry Brown, in tow. I had just arrived at the office, with a full day of work already scheduled, but had to drop everything when Steve asked me to brief Brown on the Lisa and to give him a tour of the offices.

Al Mandel and I spent an hour or so with Jerry in the Lisa "Sneak Room" (a very well designed multimedia, executive briefing facility), giving him demonstrations and encouraging him to try some stuff himself. As I recall we even took apart a Lisa to show him the hardware architecture. He was trying to feign interest but his mind was off somewhere else (as it often was during his tenure as Governor).

After the show-and-tell I then took Brown on a guided tour through the Lisa facilities and we visited engineering, documentation, marketing, etc. In the past I had taken other famous people on walkthroughs and Apple employees would usually never bat an eye. We were used to this (and after all, WE were celebrities as well). However, the reaction to Jerry Brown was unlike anything I'd ever seen. People were acting almost nervous...concerned. While Jerry was talking to an engineer, someone else took me aside, looked me straight in the eye and pointed at Jerry with a look of anguish on his face. As I had no idea what he was getting at I simply replied nonchalantly, "Yeah...that's Jerry Brown". After 10-15 minutes of this Jerry had completely lost interest. I walked him out of the offices and we bid each other goodbye. But I was still very puzzled by the reaction to him.

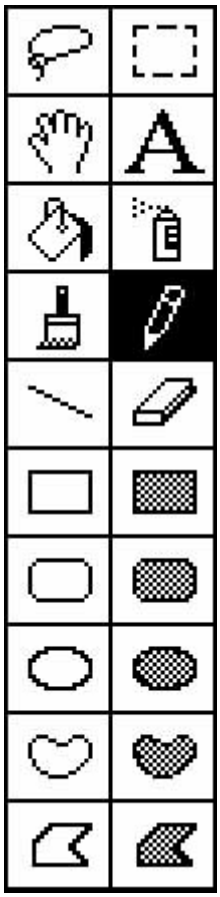
As soon as I went back in the Bandley Lobby I discovered the source of the bewilderment. The Apple rumor mill was always incredibly active and that morning (unbeknownst to me because I'd been locked up with the former Governor) the word on Bandley Drive had it that Apple was prepared to announce a new CEO that day. And Jerry's arrival at the Lisa offices nearly triggered one of the biggest Wall Street panics since Black Monday. Some people were convinced that Jerry was to be named the new CEO of Apple!

And, looking back, at that time in Apple's history this scenario was actually very plausible...

As it turns out the Apple rumor mill was correct. Later that day John Sculley was announced as the new CEO of Apple. But that's another story for another time.

# MacPaint Evolution

**Author:** Andy Hertzfeld
**Date:** June 1983
**Characters:** Steve Jobs, Bill Atkinson, Rod Perkins
**Topics:** MacPaint, Software Design, QuickDraw, Technical, Lisa
**Summary:** Bill decides to leave out a very impressive feature
**Revision:** most recent of 23



**MacPaint tool palette**

While Bill Atkinson was developing LisaGraf, the crucial, lightning fast graphics package that was the foundation of both the Lisa and Macintosh user interface (it was renamed QuickDraw for the Mac), he also worked sporadically on a simple bitmap-based drawing program for the Lisa called SketchPad. SketchPad enabled mouse-based drawing with a selection of paintbrushes and patterns, and gave Bill a quick way to test out and show off new features or improvements as they were added to LisaGraf.

In early 1983, soon after the Lisa was announced at the 1983 annual shareholders meeting, Bill switched from working on Lisa system software to writing a killer graphics application for the Macintosh. Steve Jobs thought that he should work on a structured drawing program, something like Mark Cutter's LisaDraw, but Bill thought that structured drawing was too complex, and wanted to create something that was simple, elegant and fun to use.

He began by dusting off his old SketchPad code, and getting it running on the Mac as MacSketch. SketchPad used menus to select patterns and styles to draw with, but Bill replaced them with permanent palettes at the bottom of the screen and added another large, prominent palette on the left, containing a variety of drawing tools. More tools would be added over time, but the basic structure of MacPaint was there from its earliest stages.

The first big advance that Bill worked on was eliminating flicker. As you dragged a shape or image across the screen, it had to be erased from its old position before being drawn in the new one, which caused a distracting flicker as the video sometimes displayed interim, partially rendered states. Bill completely eliminated flicker by composing everything in an offscreen memory buffer, which was transferred to the screen in one fell swoop, so the interim states were never visible.

In fact, despite the Macintosh's limited memory, he used two offscreen buffers, each the size of the document window, with one containing the current pixels of the document, and the other containing the pixels of its previous state, before the most recent operation. This made it very easy to implement undo, by just copying the old buffer to the new. It also enabled fast drawing, because it was very easy to access the original state of the document in the second buffer as an object was being modified.

After getting basic updating working well, Bill began to add more tools to the palette. He added a rectangular selection tool, to allow the user to perform operations on a subset of the document. The selection rectangle was depicted by inverting the pixels beneath it, which worked well enough over solid areas, but was confusing and hard to see when the underlying image was complex, like a digitized picture. Bill knew he had to find a better way to do it.

**the inspiration for marching ants**

He was thinking about the selection problem one evening when he went to dinner at an old beer and hamburger joint in Los Gatos, the kind of place where decades worth of initials are carved into the wooden tables. He ordered a beer, looked around the bar, and noticed a Hamm's beer sign. The beer sign featured an impressive animated waterfall, with the water seeming to flow down the waterfall into the lake. Bill figured out the animation was accomplished by a mask layer moving beneath the surface of the sign that varied which portion of the image was visible below.it.

He suddenly realized that an animated border, animated with a technique similar to the waterfall, could solve the graphical selection problem because it would always be easy to spot no matter what the background. He raced back home and implemented the animation by using an alternating sequence of patterns that produced the illusion of continuous motion, just like the beer sign. It looked great, but he didn't know how to refer to it. A couple of days later, he showed it to Rod Perkins on the Lisa applications team, who told Bill the effect resembled "marching ants," which Bill liked, so that's what he called it.

Around April 1983, Bill changed the name of the program from MacSketch to MacPaint. He began to hit his stride, and started to add new features to MacPaint on a daily basis. One of the most important was "Fat Bits", a mode which magnified a small section of the document by a factor of 8, allowing the user to easily manipulate individual pixels. It was implemented by scaling the offscreen buffer as it was transferred to the screen, so all of the other tools and effects kept working in Fat Bits mode.

Another key improvement came when Bill implemented the paint bucket tool, which allowed the user to fill an entire area of the screen with the selected pattern with a single click. It used a "seed fill" algorithm to find all the neighboring pixels of the matching color, which was difficult to implement because of our stringent memory constraints. Bill ended up implementing a few different seed fill algorithms before he settled on one that was both fast and memory efficient enough. The seed fill routine eventually migrated from MacPaint into QuickDraw as part of the 128K ROM.

Rectangular selection was useful, but often you needed to manipulate an arbitrarily shaped area of an image, so Bill came up with another selection tool he called the "lasso", because it contracted around its target. It's hard to control the mouse with pixel-level precision, so the lasso tool allowed you to roughly outline the area of interest, and it would automatically tighten, skipping white pixels to cinch tightly around the target. The lasso sort of fell out of doing the paint bucket, since the seed fill routine was exactly what was needed to cinch the lasso properly.

The lasso allowed an arbitrary area of pixels to be selected, so it required a third offscreen buffer the size of the document window, that contained a mask to indicate which pixels were part of the selection. The mask buffer allowed fast graphical operations on large, irregular areas, such as dragging an image around the screen, or making the lasso selection shimmer with the marching ants effect. The first two offscreen buffers were allocated statically, but the mask buffer needed to be deallocated when it wasn't being used, so its memory could be used for other purposes.

MacPaint was good at drawing text, allowing the user to specify characters at any position, with any font, size or style. But once the text was instantiated, it just became pixels like everything else; you couldn't go back and edit it as text. In June 1983, Bill thought he could do something about that.

Bill decided to try to turn pixels back into characters when you selected them with the text tool. He wrote a lot of elaborate code, probably as much as for any other MacPaint feature. First, he wrote assembly language routines to isolate the bounding box of each character in the selected range. Then he computed a checksum of the pixels within each bounding box, and compared them to a pre-computed table that was made for each known font, only having to perform the full, detailed comparison if the checksum matched.

Bill got his character recognition routines working well, and it seemed like magic, if you were used to the earlier MacPaint, to be able to recover and edit previously placed text. It wasn't perfect, because it would fail to recognize a character if a single dot was out of place, but it was still very useful. Everyone loved the feature, and congratulated Bill for pulling off another miracle.

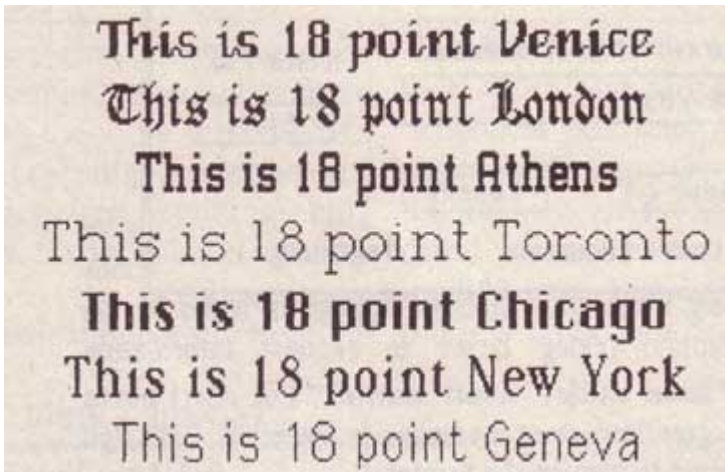
I was surprised a few days later when Bill told me that he decided to remove the character recognition feature from MacPaint. He was afraid that if he left it in, people would actually use it a lot, and MacPaint would be regarded as an inadequate word processor instead of a great drawing program. It was probably the right decision, although I didn't think so at the time. I was amazed that he was able to detach himself from all the effort that he put into creating the discarded feature; I know that I probably wouldn't have been able to do the same.

MacPaint was essentially finished by October 1983, long before our other key applications. The last part of finishing MacPaint involved dealing with out of memory problems, since it was really pushing the limits of the 128K Macintosh by using the three, large offscreen buffers. At the worst case, there was only about 100 bytes free in MacPaint's heap. Most of the bugs we encountered when running MacPaint turned out to be bugs in the underlying system, which were exposed by running so close to the edge of available memory.

It's interesting to note that MacPaint was a rather small program by today's standards, but I guess that it had to be to run in the Mac's one eighth of a megabyte of memory. The finished MacPaint consisted of 5,804 lines of Pascal code, augmented by another 2,738 lines of assembly language, which compiled into less than .05 megabytes of executable code.

# World Class Cities

**Author:** Susan Kare
**Date:** August 1983
**Characters:** Steve Jobs, Bill Atkinson, Andy Hertzfeld, Susan Kare
**Topics:** Software Design, Personality
**Summary:** We needed to come up with a naming scheme for our fonts
**Revision:** most recent of 12




**here's a sampler of some original Mac fonts**

Landing in the Macintosh group as a bitmap graphic designer was a lucky break for me, and one interesting part of the job was designing screen fonts. It was especially enjoyable because the Macintosh was able to display proportional typefaces, leaving behind the tyranny of monospace alphabets with their narrow m's and wide i's.

The first Macintosh font was designed to be a bold system font with no jagged diagonals, and was originally called "Elefont". There were going to be lots of fonts, so we were looking for a set of attractive, related names. Andy Hertzfeld and I had met in high school in suburban Philadelphia, so we started naming the other fonts after stops on the Paoli Local commuter train: Overbrook, Merion, Ardmore, and Rosemont. (Ransom was the only one that broke that convention; it was a font of mismatched letters intended to evoke messages from kidnapers made from cut-out letters ).

One day Steve Jobs stopped by the software group, as he often did at the end of the day. He frowned as he looked at the font names on a menu. "What are those names?", he asked, and we explained about the Paoli Local.

"Well", he said, "cities are OK, but not little cities that nobody's ever heard of. They ought to be WORLD CLASS cities!"

So that is how Chicago (Elefont), New York, Geneva, London, San Francisco (Ransom), Toronto, and Venice (Bill Atkinson's script font) got their names.

# Stolen From Apple

**Author:** Andy Hertzfeld

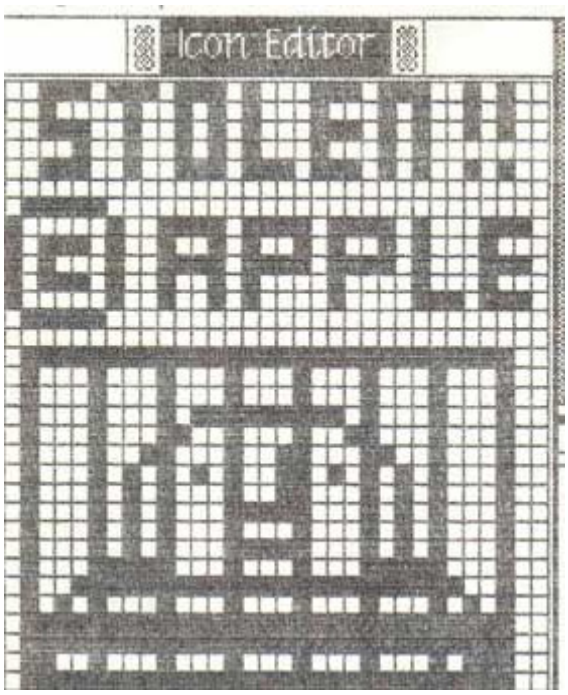
**Date:** August 1983

**Characters:** Steve Jobs, Susan Kare, Steve Capps, Andy Hertzfeld, Steve Jasik

**Topics:** Software Design

**Summary:** We put a hidden icon in the ROM

**Revision:** most recent of 2




**the secret "Stolen From Apple" icon**

In 1980, a company called Franklin Computer produced a clone of the Apple II called the Franklin Ace, designed to run the same software. They copied almost every detail of the Apple II, including all of its ROM based software and all the documentation, and sold it at a lower price than Apple. We even found a place in the manual where they forgot to change "Apple" to "Ace". Apple was infuriated, and sued Franklin. They eventually won, and forced Franklin to withdraw the Ace from the market.

Even though Apple won the case, it was pretty scary for a while, and it wasn't clear until the end that the judge would rule in Apple's favor - Franklin argued that they had a right to copy the Apple II ROMs, since it was just a "functional mechanism" necessary for software compatibility. We anticipated that someone might try a similar trick with the Macintosh someday. If they were clever enough (which Franklin wasn't), they could disguise the code (say by systematically permuting some registers) so it wouldn't look that similar at the binary level. We thought that we better take some precautions.

Steve decided that if a company copied the Mac ROM into their computer, he would like to be able to do a demo during the trial, where he could type a few keystokes into an unmodified infringing machine, and have a large "Stolen From Apple" icon appear on its screen. The routines and data to accomplish that would have to be incorporated into our ROM in a stealthy fashion, so the cloners wouldn't know how to find or remove it.

It was tricky enough to be a fun project. Susan designed a nice "Stolen from Apple" icon, featuring prison bars. Steve Capps had recently come up with a simple scheme for compressing ROM-based icons to save space, so we compressed the icon using his technique, which not only reduced the overhead but also made it much harder to detect the icon. Finally, we wrote a tiny routine to decompress the icon, scale it up and display it on the screen. We hid it in the middle of some data tables, so it would be hard to spot when disassembling the ROM.

All you had to do to invoke it is enter the debugger and type a 6 digit hexadecimal address followed by a "G", which meant execute the routine at that address. We demoed it for Steve and he liked it. We were kind of hoping someone would copy the ROM just so we could show off our foresight.

As far as I know, no one ever did copy the ROM in a commercial project, so it wasn't really necessary, but it did create some intrigue for a while. We let it slip that there was a "stolen from Apple" icon hidden in there somewhere, partially to deter people from copying the ROM. At least one hacker became moderately obsessed with trying to find it.

Steve Jasik was the author of the MacNosy disassembler/debugger, which could be used to create pseudo-source for the ROM. He found out about the "stolen from Apple" icon pretty early on, and became determined to isolate it. He lived in Palo Alto, so I would occasionally bump into him, and he would ask me for hints or tell me his latest theory about how it was concealed, which was invariably wrong.

This went on for two or three years, before he finally cracked it: I ran into him and he had it nailed, telling me about the compressed icon and the address of the display routine. I congratulated him, but was never sure if he figured it out himself or if someone with access to the source code told him.

# Pirate Flag

**Author:** Andy Hertzfeld
**Date:** August 1983
**Characters:** Steve Capps, Susan Kare, Steve Jobs
**Topics:** Apple Spirit, Buildings, Lisa Rivalry, Retreats
**Summary:** The Mac Team hoists a pirate flag
**Revision:** most recent of 5

**The pirate flag, as posed for Fortune Magazine in 1984.**

The Mac team held another off-site retreat in Carmel in January 1983, just after the Lisa introduction (see Credit Where Due) . Steve Jobs began the retreat with three "Sayings from Chairman Jobs", intended to inspire the team and set the tone for the meeting. The sayings were:

1. Real artists ship.
2. It's better to be a pirate than join the navy.
3. Mac in a book by 1986.

I think the "pirates" remark addressed the feeling among some of the earlier team members that the Mac group was getting too large and bureaucratic. We had started out as a rebellious skunkworks, much like Apple itself, and Steve wanted us to preserve our original spirit even as we were growing more like the Navy every day.

In fact, we were growing so fast that we needed to move again. In August of 1983, we moved across the street to a larger building that was unimaginatively designated "Bandley 3". I had worked there before, in 1980, when Apple had initially built it to house the original engineering organization. But now it was to be the new home of the newly christened "Macintosh Division", over 80 employees strong.

The building looked pretty much like every other Apple building, so we wanted to do something to make it look like we belonged there. Steve Capps, the heroic programmer who had switched over from the Lisa team just in time for the January retreat, had a flash of inspiration: if the Mac team was a band of pirates, the building should fly a pirate flag.

A few days before we moved into the new building, Capps bought some black cloth and sewed it into a flag. He asked Susan Kare to paint a big skull and crossbones in white at the center. The final touch was the requisite eye-patch, rendered by a large, rainbow-colored Apple logo decal. We wanted to have the flag flying over the building early Monday morning, the first day of occupancy, so the plan was to install it late Sunday evening.

Capps had already made a few exploratory forays onto the roof during the weekend while a few of us looked out for guards on the ground. At first, he thought he could just drape the flag on the roof, but that proved impractical as it was too hard to see, especially when the wind curled it up. After a bit of searching, he found a thin metal pole among the remaining construction materials still scattered inside the building, that was suitable to serve as a flag pole.

Finally, on Sunday night around 10pm, it was time to hoist the Jolly Roger. Capps climbed onto the roof while we stood guard below. He wasn't sure how he would attach the flag, and didn't have many tools with him. He scoured the surface of the roof and found three or four long, rusty nails, which he was able to use to secure the flag pole to a groove in the roof, ready to greet the Mac team members as they entered the new building the next morning.

We weren't sure how everyone would react to the flag, especially Steve Jobs, but Steve and almost everyone else loved it, so it became a permanent fixture of the building. It usually made me smile when I caught a glimpse of it as I came to work in the morning.

The flag waved proudly over Bandley 3 for about a month or two, but one morning in late September or early October, I noticed that it was gone. It turns out that the Lisa team, with whom we had a mostly friendly rivalry, decided it would be fun to steal the flag for themselves. I think they sent us a ransom note or something, so a few of us stormed over to the Lisa building to retrieve it, which we accomplished, although Capps had to wrestle it from the grasp of one of the secretaries, who was hiding it in her desk.

The flag continued to fly over Bandley 3 for more than a year. I think it was even photographed for a magazine or two during the Mac introduction. But suddenly one day it was missing again, and I'm not sure if anyone knows what happened to it. It would be a great artifact for the Computer History Museum if it ever turns up.

# Swedish Campground

**Author:** Andy Hertzfeld
**Date:** August 1983
**Characters:** Susan Kare, Steve Jobs
**Topics:** Software Design
**Summary:** How we got the unusual symbol used for the menu command key
**Revision:** most recent of 1

We thought it was important for the user to be able to invoke every menu command directly from the keyboard, so we added a special key to the keyboard to invoke menu commands, just like our predecessor, Lisa. We called it the "Apple key"; when pressed in combination with another key, it selected the corresponding menu command. We displayed a little Apple logo on the right side of every menu item with a keyboard command, to associate the key with the command.

One day, late in the afternoon, Steve Jobs burst into the software fishbowl area in Bandley III, upset about something. This was not unusual. I think he had just seen MacDraw for the first time, which had longer menus than our other applications.

"There are too many Apples on the screen! It's ridiculous! We're taking the Apple logo in vain! We've got to stop doing that!"

After we told him that we had to display the command key symbol with each item that had one, he told us that we better find a different symbol to use instead of the Apple logo, and, because it affected both the manuals and the keyboard hardware, we only had a few days to come up with something else.

It's difficult to come up with a small icon that means "command", and we didn't think of anything right away. Our bitmap artist Susan Kare had a comprehensive international symbol dictionary and she leafed through it, looking for an appropriate symbol that was distinctive, attractive and had at least something to do with the concept of a menu command.

Finally she came across a floral symbol that was used in Sweden to indicate an interesting feature or attraction in a campground. She rendered a 16 x 16 bitmap of the little symbol and showed it to the rest of the team, and everybody liked it. Twenty years later, even in OS X, the Macintosh still has a little bit of a Swedish campground in it.

# Quick, Hide In This Closet!

**Author:** Andy Hertzfeld
**Date:** August 1983
**Characters:** George Crow, Bob Belleville, Larry Kenyon, Steve Jobs, Rod Holt, Rich Williams, Hide Kamoto
**Topics:** Hardware Design, Management, Lisa
**Summary:** Steve forbid us to work with Sony
**Revision:** most recent of 4

**A Twiggy diskette**

In 1980, Apple reorganized again, splitting off a new "Disk Division" headed by John Vennard, responsible for developing a hard disk code-named "Pippin" and a next generation floppy disk code-named "Twiggy". Both were intended to be used first by the Lisa project, and eventually across Apple's entire product line. At Rod Holt's request, I had written some early diagnostics for Twiggy using an Apple II, but I felt lucky that they asked Rich Williams instead of me to transfer to the disk division as their software guy, since focusing exclusively on disks seemed pretty limiting.

Woz's Apple II floppy disk design was way ahead of the rest of the industry, so Apple felt confident that it could continue to innovate to extend its lead. Twiggy was a fairly ambitious project, more than quadrupling the capacity of standard floppy disks by doubling the data rate (which required higher density media) and employing other innovative tricks like motor speed control, which slowed down the disk rotation speed on the outer tracks to cram more data on them.

The Lisa was designed to include two built-in Twiggy drives, so it made sense for the Macintosh to use Twiggy as well. Twiggy used a Woz-style disk controller, which created a problem for the Lisa designers, since that required exact timing from the microprocessor and therefore couldn't tolerate interrupts, which was perhaps OK for a simple system like the Apple II but was unacceptable for a more sophisticated system like Lisa. Instead, the Lisa hardware designers (Paul Baker, Bob Paratore and others) solved the problem by including a little Apple II, with its own memory and microprocessor (but clocked twice as fast), inside the Lisa to control the Twiggy drives.

The Lisa also supported an optional, external hard drive through a built-in parallel port. As the Twiggy designers encountered unexpected difficulties in achieving an acceptable error rate, Lisa came to rely on the hard drive instead. The Twiggy drive was also slower than expected, because of the high error rate as well as the way the variable motor speed trick increased seek times, since you had to wait for the speed change to stabilize. Besides, the Lisa operating system designers were used to working on systems that swapped memory from disk, which wasn't really feasible to pull off at floppy disk speeds. Soon, the hard disk became mandatory, upping the minimum price of a Lisa by more than a thousand dollars.

Lisa was announced to great fanfare in January 1983, but it still wasn't ready to ship. There were problems in a number of areas, but the biggest one was the low yield of the Twiggy drives, whose high error rate greatly limited production. Finally, Lisas were shipped to customers in June 1983, even though there continued to be production and reliability problems with the disk drives.

Meanwhile, the Mac team was beginning to panic. We were using a single Twiggy drive as our floppy disk, and we didn't have a hard disk to fall back on. It looked like the Twiggy drive was never going to be reliable or cost effective enough for the Macintosh, but we were stuck without an alternative. If we couldn't find a suitable replacement quickly enough, we'd have to slip the entire project indefinitely.

Fortunately, we were aware of Sony's new 3.5 inch drive that they started to ship in the spring of 1983 through Hewlett-Packard, their development partner. George Crow, the analog engineer who designed the Mac's analog board, had come from HP prior to working at Apple and was sold on the superiority of the Sony drives. He procured a drive from his friends at HP and proposed to Bob Belleville that we figure out how to interface it to the Mac as soon as possible, while we negotiate a deal with Sony.

The Sony drive looked really sweet, especially when compared to the Twiggy. It used the same data rate as Twiggy, but on smaller disks that could fit in a shirt pocket. Best of all, the media was encased in a hard plastic shell, making it much less fragile and more convenient to handle.

Steve Jobs was finally ready to acknowledge reality and give up on the Twiggy drive. When he saw the Sony drive he loved it, and immediately wanted to adapt it for the Mac. But instead of doing the obvious thing and striking a deal with Sony, Steve decided that Apple should take what we learned from Twiggy and engineer our own version of a 3.5" drive, working with our Japanese manufacturing partner Alps Electronics, who manufactured the Apple II floppy drives at a very low cost.

This seemed like suicide to George Crow and Bob Belleville. The Mac was supposed to ship in less than seven months, and it was preposterous to think that we could get a 3.5" drive into production by then, if we could do it at all, given the disk division's dismal track record. But Steve was convinced that we should do our own drive, and told Bob to cease all work on the Sony drive. He instructed Rod Holt, Bob and George to fly to Japan to meet with Alps to initiate a crash project to develop a workable 3.5 inch drive.

Bob and George grudgingly went along with the Alps program, but they were certain that the team would discover that we couldn't pull it off in the alloted time frame. They hatched an alternative plan to continue to work with Sony surreptitiously, against Steve's wishes. Larry Kenyon was given a Sony drive to interface to the Mac, but he was told to keep it hidden, especially from Steve. Bob and George also arranged meetings with Sony, to discuss the customizations that Apple desired and to hammer out the beginnings of a business deal.

This dual strategy entailed frequent meetings with both Alps and Sony, with the added burden of keeping the Sony meetings secret from Steve. It wasn't that hard to do in Japan, since Steve didn't come along, but it got a little awkward when Sony employees had to visit Cupertino. Sony sent a young engineer named Hide Kamoto to work with Larry Kenyon to spec out the modifications that we required. He was sitting in Larry's cubicle with George Crow when we suddenly heard Steve Jobs's voice as he unexpectedly strode into the software area.

George knew that Steve would wonder who Kamoto-san was if he saw him. Thinking quickly, he immediately tapped Kamoto-san on his shoulder, and spoke hurriedly, pointing at the nearby janitorial closet. "Dozo, quick, hide in this closet. Please! Now!"

Kamoto-san looked confused but he got up from his seat and hurried into the dark janitorial closet. He had to stay there for five minutes or so until Steve departed and the coast was clear.

George and Larry apologized to Kamoto-san for their unusual request. "No problem. ", he replied, "But American business practices, they are very strange. Very strange."

As predicted, a few weeks later the Alps team came back with an eighteen month estimate for getting their drive into production, and we had to abandon the project. When Bob Belleville revealed that he and George had kept the Sony alternative alive, Steve swallowed his pride and thanked them for disobeying him and doing the right thing. The Sony drives eventually worked out great, and it's hard to imagine what the Mac would have been like without them today.

# Saving Lives

**Author:** Andy Hertzfeld
**Date:** August 1983
**Characters:** Steve Jobs, Larry Kenyon
**Topics:** Software Design, Inspiration
**Summary:** Steve wants us to make the Macintosh boot faster
**Revision:** most recent of 3

We always thought of the Macintosh as a fast computer, since its 68000 microprocessor was effectively 10 times faster than an Apple II, but our Achilles heel was the floppy disk. We had limited RAM, so it was often necessary to load data from the floppy, but there we were no faster than an Apple II. Once we had real applications going, it was clear the floppy disk was going to be a significant bottleneck.

One of the things that bothered Steve Jobs the most was the time that it took to boot when the Mac was first powered on. It could take a couple of minutes, or even more, to test memory, initialize the operating system, and load the Finder. One afternoon, Steve came up with an original way to motivate us to make it faster.

Larry Kenyon was the engineer working on the disk driver and file system. Steve came into his cubicle and started to exhort him. "The Macintosh boots too slowly. You've got to make it faster!"

Larry started to explain about some of the places where he thought that he could improve things, but Steve wasn't interested. He continued, "You know, I've been thinking about it. How many people are going to be using the Macintosh? A million? No, more than that. In a few years, I bet five million people will be booting up their Macintoshes at least once a day."

"Well, let's say you can shave 10 seconds off of the boot time. Multiply that by five million users and thats 50 million seconds, every single day. Over a year, that's probably dozens of lifetimes. So if you make it boot ten seconds faster, you've saved a dozen lives. That's really worth it, don't you think?"

We were pretty motivated to make the software go as fast as we could anyway, so I'm not sure if this pitch had much effect, but we thought it was pretty humorous, and we did manage to shave more than ten seconds off the boot time over the next couple of months.

# How to Hire Insanely Great Employees

**Author:** Tom Zito
**Date:** September 1983
**Characters:** Steve Jobs
**Topics:** Apple Spirit, Management, Recruiting
**Summary:** Steve gives the troops a pep talk
**Revision:** most recent of 5

I first met Steve Jobs in 1978 when, as a reporter for The Washington Post, I had come to the Valley to cover the technology business. Steve not only charmed me into writing a profile about him (and a year later, a cover story for Newsweek), but also charmed me into an Apple II and out of a $2,500 check to pay for it -- which at the time was more than a month's salary for me. I found the thing fairly useless, other than as an expression of how great Steve was as a salesman. A decent word processing program (not to mention VisiCalc) had yet to be written, and I wound up justifying the investment by teaching myself Basic.

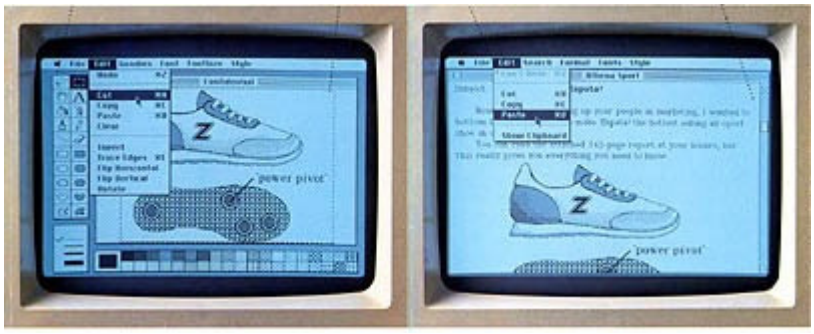
By 1983 I was working on a book about the birth of the personal computer industry, and Steve had granted me carte blanche to wander around Bandley 3 and stay current on the Mac's development. One day in September, in a conference room populated with about 25 members of the Mac team, Steve was lecturing on how to hire.

"A players hire A players," he said. "B players hire C players. Do you get it?"

Apparently not. Somebody in the back of the room raised his hand and asked, "so how do you hire more B players?"

# Cut, Paste and Crash

**Author:** Andy Hertzfeld
**Date:** September 1983
**Characters:** Larry Kenyon, Susan Kare, Andy Hertzfeld
**Topics:** Technical, Software Design, Bugs
**Summary:** We uncover a particularly bad bug shortly after ROM freeze
**Revision:** most recent of 10



**The sneaker was used to demo cut and paste**

One of the last parts of the Macintosh system software to be finished before freezing the ROM in September 1983 was the "clipboard manager", which was the code responsible for facilitating cutting and pasting information between applications. The clipboard manager provided some simple calls to access and manipulate the "clipboard", a memory buffer that held the last piece of data that was cut or copied. The trickiest part of the clipboard manager was the way it managed memory when the user quit an application.

The clipboard buffer was typically kept in the primary memory area that was available to an application, known as the "application heap". But when an application terminated, its application heap was deallocated, before a brand new heap was allocated for the incoming application. The clipboard manager had to take special measures to preserve the clipboard during this interregnum, when no application heap was available to hold it.

The first thing we thought of was to write the clipboard to disk while we were between applications, which worked but was problematic because the Macintosh didn't have a fixed disk, so there might not be a floppy in the drive, or perhaps it was write protected. We also tried to copy the clipboard into the system heap, which persisted between applications, but the clipboard was often too big to fit there. Ultimately, we solved the problem by copying the clipboard to the stack, which was a large area in high memory used to hold application variables and run-time state like return addresses, where it stayed until it was moved into the application heap of the newly launched application.

A few weeks after the ROM was frozen, Apple put together a sales "roll-out" presentation to introduce the sales team to all things Macintosh, in order to prepare them to start selling the machine a few months later. There was lots of the usual sales hoopla, but the clear highlight of the event was the first-person training on the machine itself, where each salesperson would get to try out a Macintosh for the first time. A teacher guided them through a simple demo which let them learn to use the mouse and then try out MacWrite, MacPaint and the Finder.

The climax of the demo had them loading a MacPaint document, selecting an area and copying it to the clipboard, and then launching MacWrite and pasting the image into a MacWrite document, easily mixing text and graphics, which was very impressive back in those days. Susan Kare had drawn a terrific, detailed rendering of a sneaker for the Macintosh brochure, which was used as the graphic for the demo. They were supposed to select and copy the sneaker and then paste it into MacWrite.

I was invited to go to the sales presentation, but I couldn't, because I was still working frantically with the rest of the software team to finish the software in time to ship the following January. I was busy working in the afternoon when I received a panicked phone call from one of the sales guys involved with the presentation.

"We've got a problem!", he told me. "The sneaker demo is crashing. Not all the time, though, sometimes it works and sometimes it doesn't. It's impossible to predict when it will happen, but when it does, the crash is really bad - as soon as you quit MacPaint, the screen goes crazy and then it reboots. We're going to have to stop doing the cut and paste part of the demo unless you get us a fix soon."

I got off the phone and thought about it. I was describing the problem to someone when I suddenly stopped in mid-sentence, because I realized what was going on. After quitting the application, the clipboard manager would copy the clipboard to the stack. It subtracted the size of the clipboard from the stack pointer, and then moved the memory into the allocated space. I realized that around half of the time, the size of the clipboard would be an odd number, which would end up setting the stack pointer to an odd number, which was a perversely horrible thing to do. The Mac's 68000 microprocessor couldn't fetch memory from an odd address; it generated an error if you tried. But the stack is used for the basic operation of the processor, including error handling. An odd stack pointer would essentially drive the 68000 crazy, recursively faulting as it tried to handle exceptions. It worked around half of the time because statistically the clipboard size would be an even number around half of the time. But the other half would cause the flamboyant crash that was being described.

Once I understood what was going on, it was easy to fix by rounding up the clipboard size to an even number. Unfortunately, the errant code in question was in ROM, which was already frozen in immutable silicon. This was the first major bug that I knew of in the ROM and I wondered if we were going to have to spin another version. But Larry Kenyon had already figured out a sneaky technique to fix ROM bugs, by patching system traps. We had always figured to replace entire buggy system calls in that fashion, but Larry thought of a finer grain way to do it, by patching the first trap called before or after the problem area. Basically, we could grab control any time the system was invoked, and then add code to fix problems. I used Larry's technique to devise a patch that fixed the odd stack problem, and then he helped me incorporate it into the System file, so it was loaded when the system booted.

We made a floppy containing the new System file, which was flown out to the sales presentation with a sales manager who was leaving the next day. But it took a while for the fixed System to proliferate to everyone, so for the next few weeks I had to brace myself every time that I saw someone about to cut and paste between applications.

# The Puzzle

**Author:** Andy Hertzfeld
**Date:** September 1983
**Characters:** Andy Hertzfeld, Jerome Coonen
**Topics:** Software Design, Management
**Summary:** The puzzle desk accessory became controversial
**Revision:** most recent of 1

**the Puzzle desk accessory**

The original Macintosh could only run one real application at a time, but it could also concurrently run little programs called "desk accessories" that shared memory with the main application. Like the system itself, most of the desk accessories were written in 68000 assembly language, but in the fall of 1982, I decided to write one in Pascal as a proof of concept and to show developers how to do it, by writing a tiny adaptor that the Pascal code linked with.

Desk accessories were usually utilities, like the calculator or the alarm clock, but I thought that we should also have a game or two, to show that the Macintosh was fun, too. I decided to write a "15 number puzzle", where there are fifteen numbered tiles in a four by four space that must be arranged in sequential order. If you click on a tile next to the empty space, it slides into that space. It was a fun way to waste time and build up your mouse coordination.

Since the number puzzle was written in Pascal, it had to link with the Pascal run-time, which dragged in lots of extra code that wasn't used. This made the Puzzle over 6K bytes long, even though most of it was just the run-time.

By the fall of 1983, it was time to make decisions about what to include in the shipping product. We had shown the Mac to a number of industry analysts, and while most were enthusiastic, some didn't really get the graphical user interface and thought it was "game-like", not suitable for serious computing. This made some of the Macintosh marketing folks a bit leery about the more whimsical aspects of the design, and the puzzle, being an actual game, became somewhat controversial.

Jerome Coonen, the software manager, came by my cubicle one morning to tell me that they decided not to ship the puzzle, partially because of the game-like perception, but mostly because it was just too big. Applications were very tight on RAM, and the puzzle was one of the biggest desk accessories because it was written in Pascal. At over 6K bytes, it also ate into the available disk space.

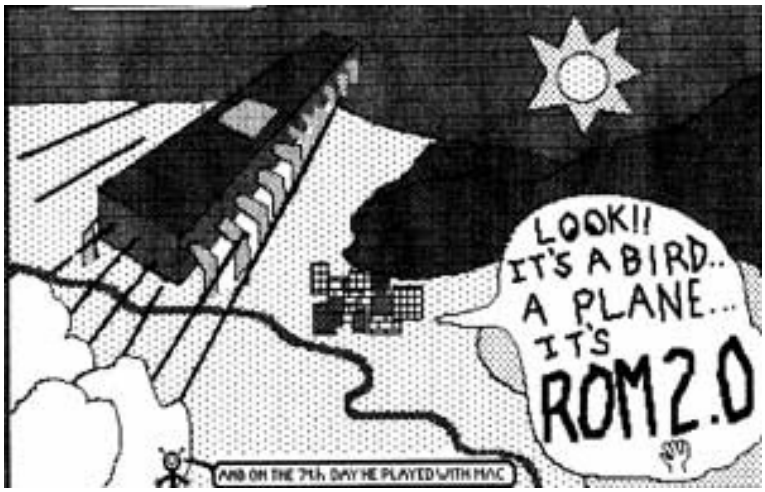
I liked the puzzle and I didn't want to capitulate to the buttoned-down, all business view of the customer, so I told Jerome, "You know, the puzzle doesn't have to be so big. I bet I could rewrite it and get it to take up less than 1K bytes. Would you keep it if I got it that small, or is it really the other issue?"

Jerome thought about it, and then told me if I could get it down to 600 bytes or so, it would be in the release. The only problem was I had to get it done over the weekend, because they had to send the manuals out to the printer soon, and there was plenty of other stuff for me to work on.

Of course, I couldn't resist a challenge like that. It only took a few hours on Saturday to recode it in assembly language, and get it down to the required 600 bytes, since it no longer had to link with the bulky Pascal code. I proudly showed it off to everyone on Monday, and it did make it into the first release, and stayed there for many years, although it was completely rewritten a couple of times, for various reasons

# We're Not Hackers!

**Author:** Andy Hertzfeld
**Date:** September 1983
**Characters:** Larry Kenyon, Andy Hertzfeld, Bill Atkinson, Burrell Smith, Steve Capps
**Topics:** Software Design, Technical, Personality
**Summary:** We were always dealing with memory limitations
**Revision:** most recent of 14




**Capps drew this with an early version of MacPaint, to celebrate a ROM release**

From the beginning, the Macintosh was conceived to be a very low cost, high volume personal computer. It was important for the design team to keep manufacturing costs as low as possible. Since memory was relatively expensive, we were always dealing with memory limitations.

One of the most clever parts of Burrell Smith's original, 68000-based digital board was the "bus transformer" logic that multiplexed the data bus, allowing him to hook up the 68000, which demanded a 16 bit data bus, to only 8 memory chips. He also included a single, byte-wide 64kbit ROM chip, so that first Macintosh, circa January 1981, had a total of 64K bytes of RAM and 8K bytes of ROM.

But as we started to get some software going on the prototype, it became increasingly clear that we didn't have enough RAM for the kind of graphic intensive applications that we wanted to build; after all, just the frame buffer for the bit-map display took up almost one third of the available memory. And furthermore, Bill Atkinson's graphics routines alone had recently exceeded the size of the 8K ROM. So, when the digital board was redesigned to incorporate the SCC chip in June 1981, Burrell added another row of 8 memory chips, doubling the RAM size to 128K, and added another ROM chip as well, doubling the ROM size to 16K. We vowed that we would fight hard to keep that the last increase (in contrast with the Lisa, whose memory requirement's were growing considerably faster than Moore's Law)

Even though the ROM size doubled to 16K, that was barely enough to contain our prototype environment if we included the graphics routines. Burrell figured he could add a third ROM chip, for 24K total. Two of the ROM chips were hooked up directly to the 68000's 16-bit bus, so code could run faster, while the third chip shared the "bus transformer" circuit with the RAMs.

We built 50 Mac prototypes in the fall of 1981, each containing 24K of ROM, burned into EPROM. Although the system fit readily in 24K, we were still worried that soon it would be an unbearably tight squeeze. Burrell never liked the inelegance of three different ROMs, either.

One day, in early 1982, Burrell returned from a meeting with a semiconductor company's sales representative really excited, almost running into my office. "OK, you say that you won't be able to fit in 24K, right? Be honest - how much will we really need?"

We always seemed to need just a little more ROM that we had available. "I think we'd definitely make it if we had 32K", I responded.

Burrell laughed. "No you won't. It's clear that won't be enough, since the software isn't close to being finished yet. But I just heard that the 256Kbit ROMs are really close, and they'll be ready if we ship in early 1983. So I can use two 256Kbit chips, connected up to the 16-bit bus, and we'll have 64K bytes of ROM. 64K! ROM is half the price per bit of RAM, so it makes sense to use as much as we can. I know you'll be asking for even more someday, but that should keep you busy for a while."

At first, 64Kbytes seemed boundless. We were already trying to write as tight code as we could, and it seemed like it would be plenty, since we weren't even using 32K yet. But sure enough, as the system came together in the spring of 1983, we were beginning to strain against the size limit.

Fortunately, we had started to use the resource manager to load objects like fonts and drivers, so we had some flexibility about keeping stuff on disk instead of the ROM. Jerome and I designed the "PACK" mechanism, where the resource manager would load code for optional packages, like the floating point routines. But code on floppy disk is much slower to load, and it also would reduce the effective size of each disk.

Even though we tried to make our code as small as possible initially, the lack of space in the ROM made us work even harder to reduce the footprint. We developed a number of unusual space saving techniques, some of which were inspired by tricks Woz used in the Apple II ROM. For example, we'd often push parameters on the stack out of order, sometimes four times in a row, because we had a value in a register that we would need later, and we didn't want to fetch it again. We knew this made the code harder to maintain, but we thought it was worth it.

As ROM freeze time approached, the entire team started to focus on code compression, to get as much code to fit in the ROM as possible. We had a few practice sessions, where everyone explained their favorite space saving techniques, and then we all plowed through the code, saving a dozen bytes here and there. Steve Capps, as usual, came up with a good one by developing a simple way of compressing the four or five icons that were built into the ROM, saving hundreds of precious bytes in the process.

Bill Atkinson didn't participate in the code crunching marathon, and wouldn't allow QuickDraw to be subjected to it, except in a few cases, because he believed that all code should be as simple and clear as possible, and thought, probably correctly, that we'd be better off without the tricks in the long run. Just before the ROM was frozen, in September 1983, he found a bug in the memory manager, that we devised a simple fix for.

I went with Bill to Larry Kenyon's cubicle, where he was maintaining the memory manager sources. Bill looked over our shoulders as we added a little code to correct the bug. But he objected when he noticed that we used one of our coding tricks (I'll explain the exact issue in a comment below) while making the fix, and began to get upset.

"Hey, that's not the right way to code. What are you guys, a bunch of hackers? I'm not sure that I want to work with a bunch of hackers."

Both Larry and I cared more about pleasing Bill than saving every possible byte or cycle, so we changed our fix to use the slower, more conservative, Bill-approved technique. We also added a comment to the instruction in the source code, to remind us why we did it the slower way in this circumstance. The comment said "We're Not Hackers!".

# 1984

**Author:** Andy Hertzfeld
**Date:** September 1983
**Characters:** Regis McKenna, Steve Hayden, Mike Murray, Steve Jobs, John Sculley, Mike Markkula, Burrell Smith, Andy Hertzfeld, Lee Clow, Jay Chiat, Brent Thomas
**Topics:** Marketing, The Launch
**Summary:** The famous 1984 commercial almost got canceled
**Revision:** most recent of 26

**the girl from the 1984 commercial**

From the very beginning, Apple always had a flair for marketing. Mike Markkula believed that a fledgling venture needed to act like a successful company if it wanted to become one, at least in terms of external perception, and Steve Jobs always insisted on the highest possible production values, even while Apple was still in the garage. The Apple II was featured in an expensive, two page spread in the September 1977 issue of Scientific American, for example, even though Apple had less than twenty employees and minimal sales at the time.

Apple's advertising agency was Chiat-Day, founded by Jay Chiat in 1968. Jay Chiat was compulsively innovative, brash and irreverent, much like an older version of Steve Jobs, and the two hit it off great when they were introduced in 1981, just before Chiat-Day acquired Regis McKenna's advertising operations. Jay and his talented team, featuring Creative Director Lee Clow and star copywriter Steve Hayden, crafted Apple's first TV commercials, recruiting talk show host Dick Cavett as a spokesperson, and created the campaign that launched the Lisa, including a TV commercial that starred Kevin Costner while he was still unknown.

Toward the end of 1982, art director Brent Thomas and Steve Hayden came up with the idea of doing an ad campaign based on the timely tagline "Why 1984 won't be like 1984". Chiat-Day shopped it around to a number of clients, including Apple, where it was proposed to be used for a print ad in the Wall Street Journal promoting the Apple II. But Apple didn't go for it, and the idea was filed away until the spring of 1983, when they met with the Mac marketing team to start working on the launch, which was scheduled for January 1984.

Steve Jobs wanted to launch the Macintosh with an inspiring commercial that was as revolutionary as the product itself. He loved the Orwellian tagline when it was presented, and he encouraged the Chiat-Day team to pursue it. Steve Hayden and Brent Thomas put together an intriguing story board, envisioning a visually striking, highly symbolic, miniature science fiction epic featuring a young female athlete who liberates the subjugated masses from totalitarian domination by throwing her sledgehammer to smash a huge screen displaying Big Brother.

Macintosh marketing manager Mike Murray and Steve Jobs loved it, but they needed to get new CEO John Sculley's approval for such a large expenditure. Sculley was a bit apprehensive (after all, the commercial hardly mentioned the Macintosh), but he gave his OK for an unprecedented production budget of over $750,000, to make the one-minute commercial.

Chiat-Day hired the best science fiction oriented director they could find, Ridley Scott, whose previous movie, Blade Runner, possessed the visionary dystopian feel they were striving for. Ridley was based in London, so they decided to shoot it there, at Shepperton Studios. Several Apple and Chiat Day executives, including Mike Murray and Steve Jobs, travelled to London for the week of filming.

By the time the Apple folks arrived, Ridley's team had assembled a cast of almost 200. To play the oppressed, downtrodden baldheaded workers, his people recruited dozens of authentic British skinheads, paying them $125 dollars a day to participate. It was harder to cast the young heroine, since most of the models who tried out had trouble spinning with the heavy sledgehammer. Luckily, one named Anya Major was an accomplished discus thrower, and she could do it faultlessly, so she got the part.

When he arrived at the studio, Mike Murray went looking for Jay Chiat, who was supposed to already be there. He found him lurking off to one side, seemingly hiding behind some scenery. Apparently, some of the skinheads were in a nasty mood, and they were looking for trouble during breaks in the filming, so Jay thought it was prudent to make sure he stayed out of their way.

While the filming was taking place in London, I got a call from someone at Chiat-Day asking if I could write an Apple II Basic program to flash impressive looking numbers and graphs on the screen, to be overlayed on the image of Big Brother. I spent an afternoon cooking something up, and I sent it off to them, although I was never sure if it was used or not.

**Big Brother on the big screen**

Lee Clow and Steve Hayden presented a rough cut of the commercial to the Apple team a few weeks later, and everyone was ecstatic. The commercial was classy, suspenseful and enigmatic, and seemed certain to garner lots of attention. It was shown for the first time at Apple's 1983 annual sales conference in Honolulu in October, with Steve preceding it with a clever rap positioning Apple as the industry's last alternative to IBM (see The Times They Are A-Changin') , to a rapturous reception, almost as apocalyptic as the commercial itself. The response was so great that Apple booked two expensive slots, for sixty seconds and thirty seconds, costing over a million dollars, to show it during Super Bowl XVIII, which was just two days before the Mac introduction.

Mike Murray and Steve Jobs screened the commercial for Apple's board of directors in December, to get final approval for the huge Superbowl expenditure. To their surprise, every outside board member seemed to despise the commercial, with Mike Markkula suggesting that Apple begin a hunt for a new ad agency. One of the board members remarked that it was the worst commercial that he had ever seen. Steve and Mike were devastated.

The chilling reception from the board compelled John Sculley to ask Chiat-Day to sell back both time slots that they had purchased. But Jay Chiat was true to form, and only sold off the thirty second slot, telling Apple that he wasn't able to get rid of the longer one at so late a date. Apple considered using the slot for a more conventional commercial, but in the end decided to take a chance on the 1984 spot.

The Mac team was told that the commercial would air early in the third quarter, at the first commercial break after the second half kick-off. Burrell and I wanted to see a real audience's reaction to the commercial more than the commercial itself (since we had already seen it plenty of times), so we watched the Superbowl at a sports bar near Stanford called the Oasis, with some other Mac team friends. The game was boring, but the bar was packed, and the commercial looked great when it aired at the designated time. We thought we heard a small murmur in the bar after the commercial, but it was hard to tell if there was any significant reaction.

That evening, we were surprised to see the commercial run again on the evening news shows. Apparently, it made such a big impression on lots of viewers that, coupled with the fact that it only was supposed to run once, it became a news item itself, as well as increasing expectations for the upcoming launch. It ran dozens of times on news shows in the next couple of days, gathering Apple over five million dollars worth of free publicity.

A week after the Macintosh launch, Apple held its January board meeting. The Macintosh executive staff was invited to attend, not knowing what to expect. When the Mac people entered the room, everyone on the board rose and gave them a standing ovation, acknowledging that they were wrong about the commercial and congratulating the team for pulling off a fantastic launch.

Chiat-Day wanted the commercial to qualify for upcoming advertising awards, so they ran it once at 1 AM at a small television station in Twin Falls, Idaho, KMVT, on December 15, 1983. And sure enough it won just about every possible award, including best commercial of the decade. Twenty years later it's considered one of the most memorable television commercials ever made.

# Close Encounters of the Steve Kind

**Author:** Tom Zito
**Date:** September 1983
**Characters:** Steve Jobs, Donald Knuth, Lynn Takahashi
**Topics:** Personality, Inspiration
**Summary:** Steve Jobs meets Donald Knuth
**Revision:** most recent of 7

Steve had managed to get Don Knuth, the legendary Stanford professor of computer science, to give a lunchtime lecture to the Mac team. Knuth is the author of at least a dozen books, including the massive and somewhat impenetrable trilogy "The Art of Computer Programming." (For an amusing look at Knuth's heady self image, and his $2.56 reward program, see http://www-cs-faculty.stanford.edu/~knuth/books.html )

I was sitting in Steve's office when Lynn Takahashi, Steve's assistant, announced Knuth's arrival. Steve bounced out of his chair, bounded over to the door and extended a welcoming hand.

"It's a pleasure to meet you, Professor Knuth," Steve said. "I've read all of your books."

"You're full of shit," Knuth responded.

# Steve Wozniak University

**Author:** Andy Hertzfeld
**Date:** September 1983
**Characters:** Burrell Smith, Steve Wozniak
**Topics:** The Launch, Personality
**Summary:** Burrell's educational credentials
**Revision:** most recent of 1

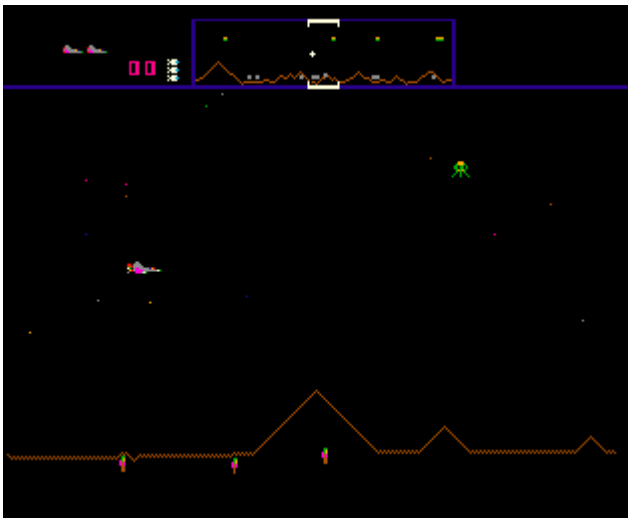
The launch of the Macintosh was finally scheduled for January 24, 1984, so we had to start preparing for it in the fall of 1983. Steve decided to anoint some of the engineers plus a few others as the official "Macintosh Design Team", and we had to be prepared to be interviewed by the press. Some magazines had lead times of more than three months, so interviews were scheduled as early as October.

As part of the preparations, the Regis McKenna press liasons (known as the Rejettes) asked each of us to fill out a questionnaire that included the standard questions like date and place of birth, degrees, etc. They were going to be used as the basis for a fact sheet that would be handed out to the press.

Burrell, who didn't have a college degree, didn't know how to answer the "College Attended" question. He thought about it for a moment and then came up with the perfect answer: "Steve Wozniak University". In some fashion, we were all graduates of Steve Wozniak University.

# Make a Mess, Clean it Up!

**Author:** Donn Denman
**Date:** September 1983
**Characters:** Burrell Smith, Donn Denman, Andy Hertzfeld
**Topics:** Personality, Recreation
**Summary:** Burrell had a unique approach to playing Defender
**Revision:** most recent of 6



**Defender was Burrell's favorite video game**

Working 90 hours a work week requires frequent, and highly effective, work breaks. In the center of Macintosh work area in Bandley 3 we had a ping pong table, a nice stereo system, and a Defender video game machine. We found that competitive play gave us a jolt of adrenaline, and a refreshed mind-set when we resumed work. We also learned a lot about our coworkers and how they excel during competition. While playing Defender one day I got some great insight into how Burrell accelerates his own learning process.

Andy, Burrell and I had a standing competition playing on the Defender machine. We'd challenge each other, in two or three player competitions, taking turns at the video game machine, and compare techniques and high scores. We were roughly equal in skill level, so as we'd take turns at the controls we could watch how the other player was doing, and have a gauge for who was ahead. This gave us opportunities to refine our skills, learn the other guy's technique, and show off our prowess.

The goal of Defender is to defend your humans from abduction by aliens. The evil green aliens drop down from the top of the screen and randomly pick up your humans, and try to bring them back up to the top of the screen. You control a ship and have to shoot the aliens, either before they grab a human, or during their rise up to the top of the screen. If an alien makes it to the top with a human, they consume him and become a vicious mutant, which attacks very aggressively. You start the game with ten humans, and if the last one dies, all the aliens become mutants, and they swarm in on your ship from all sides.

After a while, surviving the first few game levels was pretty easy, unless you had been up all night programming or something. The Defender machine was probably a pretty good objective measure of current mental capacity. "Gee, I can't even get through level 2! I guess it's time to get some sleep." Better to put in a bad performance on the defender game than mess up the current progrmming task, or start down the wrong path on some hardware design.

**Woz playing Defender in Bandley 3**

One day Burrell started doing something radical. Andy came by my cube and said "You've got to come see what Burrell's doing with Defender." "How can you innovate with a video game?" I wondered. I'd seen Burrell and Andy innovate on all kinds of things, but I couldn't image how he could somehow step outside the box of a video game - the machine controlled the flow and dictated the goals. How could you gain some control in that environment?

We started up a new competition, and when Burrell's turn came up, he did something that stunned me. He immediately shot all his humans! This was completely against the goal of the game! He didn't even go after the aliens, and when he shot the last human, they all turned to mutants and attacked him from all sides. He glanced in my direction with a grin on his face and said "Make a mess, clean it up!" and proceeded to dodge the swarm of angry mutants noisily chasing after him. "Burrell's not going to win this competition" I said to myself. "He's not going to last long with a screen full of mutants!"

Often a single mutant is enough to kill you. They move quicker, and with a different pace and pattern than the other aliens, so the normal evasive techniques don't work very well. Mutants move so quickly over small distances that they seem to just jump on top of you. Your ship is faster over the longer term, so you have to outrun them, establishing a gap, and only then do you have enough room to safely turn and fire at them.

When Burrell's next turn came up I was surprised by how long his ship survived. He'd already developed a technique for dealing with a whole mass of mutants. He would circle around them again and again, and that would gather them into a densely clumped swarm. Then, while circling, he'd fire a burst pattern across the whole swarm, not needing to aim at individuals. He was doing really well, cutting through the swarm like the Grim Reaper's scythe. Burrell was no longer attacking individual mutants, instead he was treating the whole swarm as one big target.

Burrell may have lost that game and the next few, but it wasn't too long before he was really mastering the machine. Instead of avoiding the tough situations, he'd immediately create them, and immediately start learning how to handle the worst situation imaginable. Pretty soon he would routinely handle anything the machine could throw at him.

I was beginning to see how Burrell could be so successful with everything he does. Like many high achievers, Burrell likes challenges so much that he actually seeks them out and consciously creates them. In the long run, this approach makes sense. He seems to aggressively set up challenging situations throughout his life. Then, when life throws him a curve ball, he'll swing hard, and knock it out of the park.

Why intentionally "make a mess?" So you can get really good at "cleaning up!"

# Price Fight

**Author:** Andy Hertzfeld
**Date:** October 1983
**Characters:** Steve Jobs, John Sculley, Burrell Smith
**Topics:** Marketing, Twiggy
**Summary:** We feel betrayed by the unexpectedly high price
**Revision:** most recent of 11

The Macintosh was originally intended to be a very low cost, high volume personal computer. We wanted to keep the price as low as possible, so the Mac would be affordable to ordinary individuals, and Apple could sell them by the millions. The initial target price was $500, less than half the price of an Apple II at the time, but it quickly rose to $1000 after the design team added up the cost of various components.

In early 1981, after switching from the 6809 to the more expensive 68000 microprocessor and doubling our RAM size to 128K bytes, we realized that we'd have to raise the retail price to $1500 in order for Apple to make its standard profit margin. $1500 was approximately the original price of the Apple II, and it seemed like that was about as high as you could go while still being affordable to individuals. We worked hard to keep the price from rising further, and were able to hold it at $1500 for most of the time the product was under development.

Pricing a brand new computer is tricky, because costs are highly dependent on volume: the more units of a component that you were willing to order, the lower the price per unit. But how can you predict how well a new type of computer will sell? It's literally a confidence game, and we had no shortage of that. Steve Jobs **knew** that we were going to sell Macintoshes by the millions, and he was good at convincing our suppliers to share some of the risk with us via lower initial prices, to be rewarded as volumes soared in the years ahead. For example, Steve was able to get Motorola to commit to a price of $9.00 for the 68000 microprocessor, less than a quarter of what they were currently quoting at the time.

By the summer of 1983, it was becoming clear that the disk division's Twiggy floppy disk drive wasn't going to make it, and if we weren't careful, it could drag down the Macintosh with it. We had to scurry (see Quick, Hide In This Closet!) , but we were able to replace Twiggy with the Sony 3.5 inch drive without slipping the schedule, which was better in every way except one: it cost us an extra $50 or so. When combined with a few other recent splurges, it pushed us over the top, so we grudgingly accepted that the Macintosh would have to debut for $1995.

Meanwhile, Apple hired a new CEO, John Sculley, in April 1983. John was the former CEO of Pepsi, and a world-class marketing whiz, having invented the concept of the "Pepsi Generation" and other successful promotions. He was hired by Apple mainly to apply his marketing skills to the personal computer market, and the Macintosh in particular. But big time marketing costs big time money.

In October 1983, as plans for the Macintosh launch were being finalized, and we were frantically trying to finish the software, Steve Jobs strode into the software area one evening, looking angry. "You're not going to like this," he told us, "but Sculley is insisting that we charge $2495 for the Mac instead of $1995, and use the extra money for a bigger marketing budget. He figures that the early adopters will buy it no matter what the price. He also wants more of a cushion to protect Apple II sales. But don't worry, I'm not going to let him get away with it!"

The design team was horrified. One of the main reasons that we were so passionate about the Macintosh was that we thought we were working on something that we would use ourselves, along with our friends and relatives. It was crucial that it be affordable to ordinary people. $2500 felt like a betrayal of everything that we were trying to accomplish. We worked very hard to keep the price down in every aspect of the design, and now it was being artificially inflated for reasons that didn't make sense to us. But we thought that Steve would prevail, and be able to convince John that we'd do better at the lower price.

But finally, much to our surprise and dismay, after a week or so of wrangling, Steve was the one who gave in, and the Mac was priced at $2495 at launch. Even though it sold quickly at first, soon sales bogged down, partially due to the lack of available software, but also because of the price. Even after sales picked up in 1986, with the Mac Plus and the proliferation of desktop publishing, Apple continued to overcharge for the Macintosh, preferring huge profit margins to growing their market share, which eventually led to big problems when it caught up with them in the nineties.

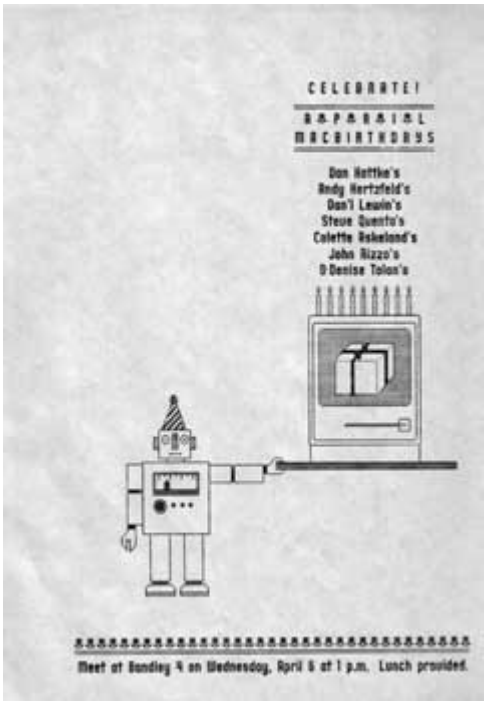
# MacPaint Gallery

**Author:** Andy Hertzfeld
**Date:** October 1983
**Characters:** Susan Kare, Bill Atkinson, Larry Kenyon, Jerome Coonen, Steve Jobs, Patti King
**Topics:** Social Life, Testing, MacPaint, Marketing
**Summary:** A gallery of Susan Kare's MacPaint art from 1983
**Revision:** most recent of 15

Bill Atkinson began writing MacPaint in February 1983, just after Susan Kare joined the Mac team to design bitmaps for fonts and icons. Susan became one of the first and most accomplished users of MacPaint, trying out new features as they were developed and using it for a wide range of practical applications.

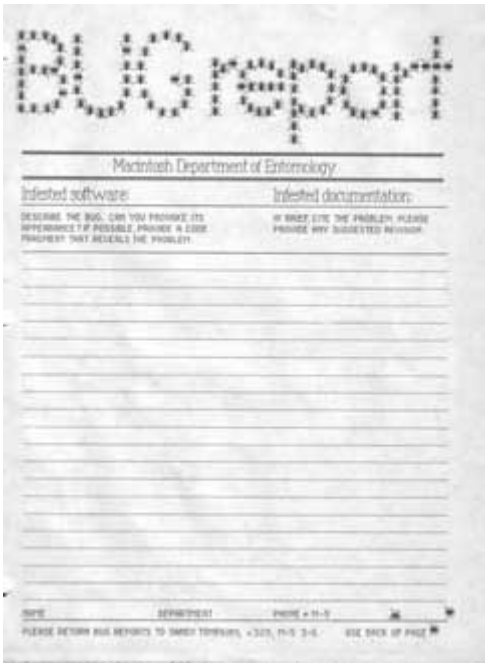
Susan kept a notebook of many of the MacPaint documents that she created as the Mac team struggled to finish the Macintosh throughout 1983. They provide an interesting glimpse of the daily life of the Mac team during that period. I'm pleased to be able to present scans her beautiful drawings here as part of Macintosh Folklore.

You can click on an image to see a larger rendition of it; use the back button to return to the story.



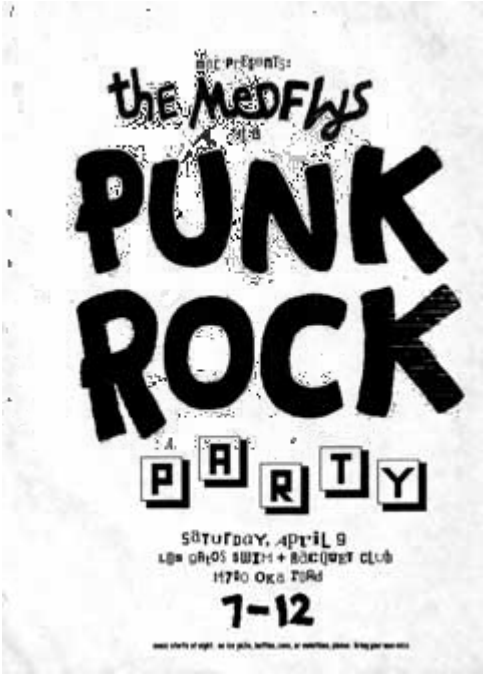
The image on the left is an annoucement of a combined birthday party for seven Mac team members whose birthday was in early April, including myself, who was terrified of turning 30 at the time. As a birthday gift, Susan made a jersey for me with a large hexadecimal number "1E" (which is 30 in base 16) on it, so I could still say I was a teenager, at least in hexadecimal.

The image on the right is for picnic held in July to celebrate the wedding of two members of the software team: programmer Larry Kenyon and librarian Patti King. Larry and Patti actually eloped in June; the picnic happened after they returned from their honeymoon.

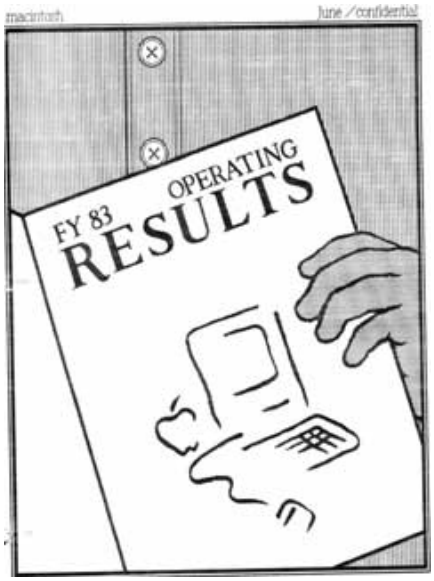



One of the most important activities during the last couple of months before shipping was software testing. We held some testing marathons, where the software team enticed employees from the rest of the division to stay late and help test the software by bribing them with dinner [story:90 Hours A Week And Loving It]. The image on the left is an announcement of another bug hunt, and the one on the right is a bug report form that was used during the testing.

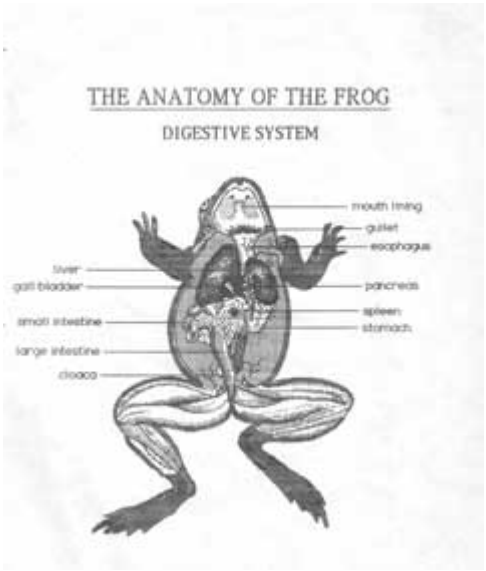
Here is an invitation to celebrate a ROM freeze at software manager Jerome Coonen's house. This wasn't the final ROM freeze, which took place in September, but the first of a series that led up to it. We had a party celebrating the final ROM freeze at Woz's house in Scott's Valley.

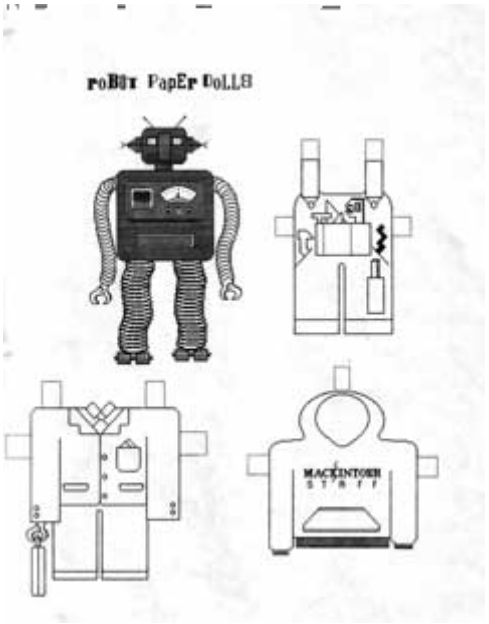


The Mac group worked hard, but we occasionally had parties thrown by Apple for the team, sometimes with unusual themes. The left most poster promoted our "Punk Party", which we were supposed to attend in "punk" attire and featured the local band "The Medflys". The poster on the right advertised a dance with music supplied by a DJ. Some of us were so disappointed with the music that he played that we drove to a nearby record store during the party to buy some better music to play there.

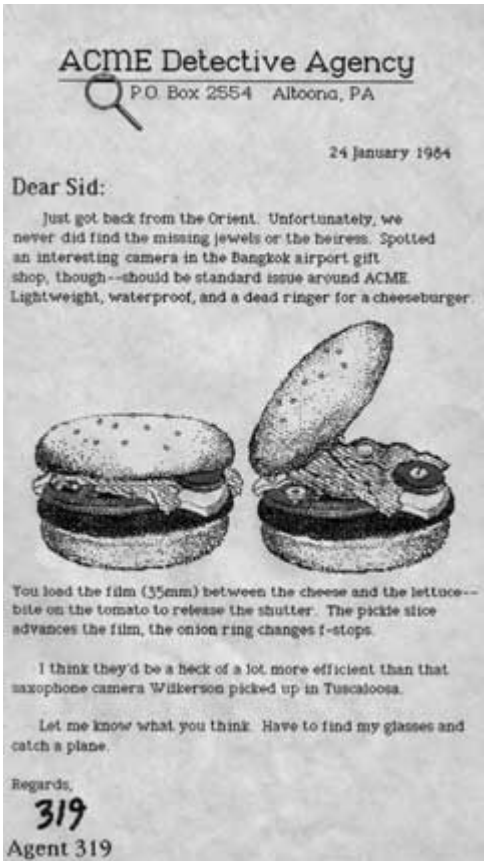


Steve Jobs was very visually oriented, and everyone knew that he reacted more favorably to material presented with nice graphics and high production values. Susan was recruited by the Finance team to use MacPaint to make attractive covers for the monthly financial reports. Here are some covers that she made for the June and July 1983 reports.



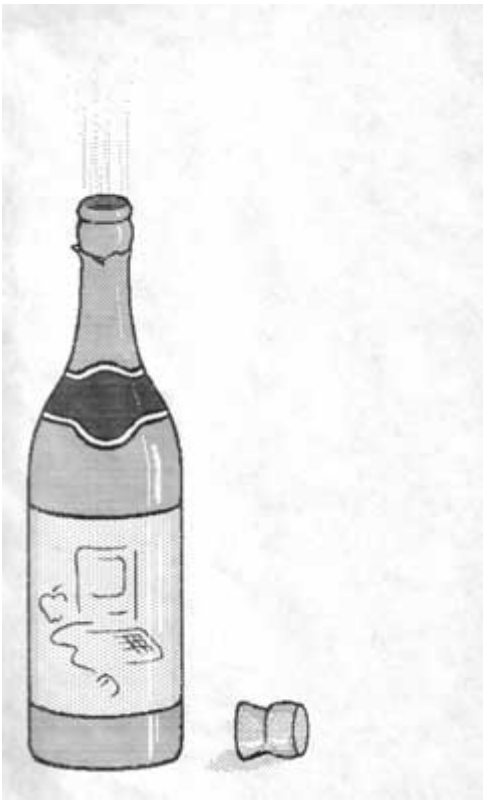




As we began preparing for the launch in the fall of 1983, Susan was requested to produce art for various marketing materials, to show off the kinds of things you could do with MacPaint. The frog on the left demonstrates how you might use MacPaint in a biology class. Susan also had a fondness for whimsical applications, as demonstrated by the one on the right.




Here are some examples that Susan created for marketing. The "Japanese Lady" on the left is a pretty famous one, since it was used in the original brochure. She started with a scan of a fine Japanese woodcut that Steve had procured. I think the detective agency letter on the right was also used in an advertisement.

Sports were another outlet to blow off steam between longs hours of working. In the spring of 1983, we managed to get facilities to install a basketball hoop in the back of Bandley 4, and usually played a half-court game every afternoon. We also had co-ed softball games once per month, which could sometimes get pretty intense.



Apple gave out brand new Macintoshes to everyone on the team right after the launch (see The Times They Are A-Changin') ; on the left is a certificate that accompanied each machine. On the right is a bottle of champagne, in honor of the launch, to complete the celebration.

# Monkey Lives

**Author:** Andy Hertzfeld
**Date:** October 1983
**Characters:** Steve Capps, Bill Atkinson
**Topics:** Software Design, Testing, Technical
**Summary:** The very first location in low memory
**Revision:** most recent of 2

The original Macintosh only had 128K bytes of RAM (that's one eighth of a megabyte), so dealing with memory management was usually the hardest part of writing both the system and applications. We allocated around 16K bytes for system use, and another 22K bytes for the 512 by 342 black and white screen, so applications were left with only 90K bytes or so. The bigger ones like MacWrite or MacPaint seemed to be bursting at the seams.

By the fall of 1983, MacWrite and MacPaint were pretty much feature complete but still needed a lot of testing, especially in low memory conditions. MacPaint needed to allocate three off-screen buffers, with each the size of the entire screen, so it was always skirting the edge of running out of memory, especially when you brought up a desk accesory, but the specific sequences that led to crashes were difficult to reproduce.

Steve Capps had been working on a "journaling" feature for the "Guided Tour" tutorial disc, where the Macintosh could demo itself by replaying back events that were recorded in a prior session. He realized that the so-called "journaling hooks" that were used to feed pre-recorded events to the system could also be the basis of a testing tool he called "The Monkey".

The Monkey was a small desk accessory that used the journaling hooks to feed random events to the current application, so the Macintosh seemed to be operated by an incredibly fast, somewhat angry monkey, banging away at the mouse and keyboard, generating clicks and drags at random positions with wild abandon. It had great potential as a testing tool, so Capps refined it to generate more semantically rich events, with a certain percentage of the events as menu commands, a certain percentage as window drags, etc.

The Monkey proved to be an excellent testing tool, and a great amusement, as well. Its manic activity was sort of hypnotic, and it was interesting to see what kind of MacPaint pictures the Monkey could draw, or if it would ever produce something intelligible in MacWrite. At first it could crash the system fairly easily, but soon we fixed the more obvious bugs. We thought it would be a good test for an application to see if it could run the Monkey all night, but usually it didn't run for more than 20 minutes, even if it didn't crash, because the Monkey would invariably select the quit command.

Bill Atkinson came up with the idea of defining a system flag called "MonkeyLives" (pronounced with a short "i" but often mispronounced with a long one), that indicated when the Monkey was running. The flag allowed MacPaint and other applications to test for the presence of the Monkey and disable the quit command while it was running, as well as other areas they wanted the Monkey to avoid. This allowed the Monkey to run all night, or even longer, driving the application through every possible situation.

We kept our system flags in an area of very low memory reserved for the system globals, starting at address 256 ($100 in hexadecimal), since the first 256 bytes were used as a scratch area. The very first slot in the system globals area, address 256, had just been freed up, so that's where we

put the MonkeyLives boolean. The Monkey itself eventually faded into relative obscurity, as the 512K Macintosh eased the memory pressure, but its memory was kept alive by the curious name of the very first value defined in the system globals area.

# The Mythical Man Year

**Author:** Andy Hertzfeld
**Date:** October 1983
**Characters:** Steve Jobs, Bill Atkinson
**Topics:** Software Design, Personality, Reality Distortion, QuickDraw
**Summary:** Steve estimates the effort that went into Quickdraw
**Revision:** most recent of 1

**Bill and Burrell on the cover of Byte**

One of our first encounters with the press was a group interview with Byte magazine in October 1983. We wanted an article to come out concurrently with the Mac intro the third week of January, and Byte had a three month lead time, so they were the first.

Byte was one of the first PC hobbyist magazines, written for a fairly technical audience of computer enthusiasts. Five or six of us were being extensively quizzed by two Byte editors, including Steve Jobs. We were talking about the Mac's graphical user interface software, and how long it took to develop.

Quickdraw, the amazing graphics package written entirely by Bill Atkinson, was at the heart of both Lisa and Macintosh. "How many man-years did it take to write QuickDraw?", the Byte magazine reporter asked Steve.

Steve turned to look at Bill. "Bill, how long did you spend writing Quickdraw?"

"Well, I worked on it on and off for four years", Bill replied.

Steve paused for a beat and then turned back to the Byte reporter. "Twenty-four man-years. We invested twenty-four man-years in QuickDraw."

Obviously, Steve figured that one Atkinson year equaled six man years, which may have been a modest estimate.

# 90 Hours A Week And Loving It!

**Burrell wearing the sweatshirt**

Most of the Macintosh software team members were between twenty and thirty years old, and with few family obligations to distract us, we were used to working long hours. We were passionate about the project and willing to more or less subordinate the rest of our lives to it, at least for a while. As pressure mounted to finish the software in time to meet our January 1984 deadline, we began to work longer and longer hours. By the fall of 1983, it wasn't unusual to find most of the software team in their cubicles on any given evening, week day or not, still tapping away at their keyboards at 11pm or even later.

The rest of the Macintosh team, which had now swelled to almost a hundred people, nearing the limit that Steve Jobs swore we would never exceed, tended to work more traditional hours, but as our deadline loomed, many of them began to stay late as well to help us test the software during evening testing marathons. Food was brought in as a majority of the team stayed late to help put the software through its paces, competing to see who could find the most bugs, of which there were still plenty, even as the weeks wore on.

**Rear View of the Sweatshirt**

Debi Coleman's finance team decided to commemorate the effort that the entire team was putting forth in the traditional Silicon Valley manner: they made a T-Shirt. Actually, to make it a little more special, they chose a high quality, gray hooded sweatshirt. Steve Jobs had recently bragged to the press that the Macintosh team was working "90 hours a week". They decided that the tag line for the sweatshirt should be "90 Hours A Week And Loving It", in honor of Steve's exaggerated assertion.

The sweatshirt featured the name Macintosh in red letters, purposefully misspelled as "Mackintosh", as it had been in a recent article, with a black squiggle crossing out the errant 'k'. The "90 Hours" tag line was emblazoned in black across the back. They were produced in time for the next testing marathon, as a reward for participating. The software team wasn't all that pleased, since we felt that we really were working that hard, but most of the other sweatshirt recipients weren't even coming close. But it was a pretty nice sweatshirt, so lots of the engineers started wearing them frequently, including Burrell Smith.

When Burrell finally quit Apple in February 1985, he continued to wear the sweatshirt almost every day, but, as soon as he returned home following his resignation, he took some masking tape and made a big 'X' across the leading '9' character, virtually obliterating it from view. He proudly displayed the updated motto, reflecting exactly how he felt. It now read "0 Hours A Week And Loving It".

# A Rich Neighbor Named Xerox

**Author:** Andy Hertzfeld
**Date:** November 1983
**Characters:** Steve Jobs, Neil Konzen, Mike Boich, Bill Gates
**Topics:** 3rd party developers, Microsoft, Personality
**Summary:** Steve confronts Bill Gates about copying the Mac
**Revision:** most recent of 14

**The Xerox Star was an influential predecessor**

When Steve Jobs recruited Microsoft to be the first third party applications software developer for the Macintosh, he was already concerned that they might try to copy our ideas into a PC-based user interface. As a condition of getting an early start at Macintosh development, Steve made Microsoft promise not to ship any software that used a mouse until at least one year after the first shipment of the Macintosh.

Microsoft's main systems programmer assigned to the Mac project was Neil Konzen, a brilliant young Apple II hacker who grew up in their backyard in the suburbs of Seattle. Neil started working at Microsoft while he was still a high school student, and single-handedly implemented the system software for their hit Z80 card that allowed the Apple II to run CP/M software

Neil loved Apple, so it was natural for Microsoft to assign him to their new, top-secret Macintosh project. He was responsible for integrating Microsoft's byte-code based interpreted environment (which actually was a copy of a system used at Xerox that favored memory efficiency over execution speed, which was appropriate for the Mac's limited memory) with the rapidly evolving Macintosh OS, so he quickly became Microsoft's expert in the technical details of the Mac system.

By the middle of 1983, Microsoft was far enough along to show us working prototypes of their spreadsheet and business graphics programs, Multiplan and Chart (they were also working on a word processor, but they neglected to mention that, since it would compete with MacWrite). I would usually talk with Neil on the phone a couple of times a week. He would sometimes request a feature that I would implement for him, or perhaps complain about the way something was done. But most of the time I would answer his various questions about the intricacies of the still evolving API.

I gradually began to notice that Neil would often ask questions about implementation details that he didn't really need to know about. In particular, he was really curious about how regions were represented and implemented, and would often detail his theories about them to me, hoping for confirmation.

Aside from intellectual curiosity, there was no reason to care about the system internals unless you were trying to implement your own version of it. I told Steve that I suspected that Microsoft was going to clone the Mac, but he wasn't that worried because he didn't think they were capable of doing a decent implementation, even with the Mac as an example.

In November 1983, we heard that Microsoft made a surprising announcement at Comdex, the industry's premier trade show, held twice a year in Las Vegas. Microsoft announced a new, mouse-based system graphical user interface environment called Windows, competing directly with an earlier environment announced by Personal Software called "Vision". They also announced a mouse-based option for Microsoft Word. When Steve Jobs found out about Windows, he went ballistic.

"Get Gates down here immediately", he fumed to Mike Boich, Mac's original evangelist who was in charge of our relationships with third party developers. "He needs to explain this, and it better be good. I want him in this room by tomorrow afternoon, or else!"

And, to my surprise, I was invited to a meeting in that conference room the next afternoon, where Bill Gates had somehow manifested, alone, surrounded by ten Apple employees. I think Steve wanted me there because I had evidence of Neil asking about the internals, but that never came up, so I was just a fascinated observer as Steve started yelling at Bill, asking him why he violated their agreement.

"You're ripping us off!", Steve shouted, raising his voice even higher. "I trusted you, and now you're stealing from us!"

But Bill Gates just stood there coolly, looking Steve directly in the eye, before starting to speak in his squeaky voice.

"Well, Steve, I think there's more than one way of looking at it. I think it's more like we both had this rich neighbor named Xerox and I broke into his house to steal the TV set and found out that you had already stolen it."

Unfortunately, it turned out that while the agreement that Microsoft signed in 1981 stipulated that they not ship mouse-based software until a year after the Mac introduction, that ended up being defined in the contract as September 1983, since in late 1981 we thought that the Mac would ship in the fall of 1982, and we foolishly didn't let the ship date float in the contract. So Microsoft was within their rights to announce Windows when they did. Apple still needed Microsoft's apps for the Macintosh, so Steve really couldn't cut them off.

Microsoft didn't manage to ship a version of Windows until almost two years later, releasing Windows 1.0 in the fall of 1985. It was pretty crude, just as Steve had predicted, with little of the Mac's thoughtful elegance. It didn't even have overlapping windows, preferring a simpler technique called "tiling". When its utter rejection became apparent a few months later, Bill Gates fired the implementation team and started a new version from scratch, led by none other than Neil Konzen.

Neil's version of Windows, released a couple of years later, was good enough that Apple filed a monumental copyright lawsuit against Microsoft in 1988, but they eventually lost on a technicality (the judge ruled that Apple inadvertently gave Microsoft a perpetual license to the Mac user interface in November 1985).

# Steve Capps Day

**Author:** Andy Hertzfeld
**Date:** December 1983
**Characters:** Jerome Coonen, Steve Capps, Bruce Horn, Larry Kenyon
**Topics:** Personality, Apple Spirit
**Summary:** We all dress up in honor of Steve Capps
**Revision:** most recent of 15

**The software team, all dressed up like Capps**

The Macintosh software finally seemed to be coming together as the fall of 1983 wore on. The ROMs containing most of the system software were finished, and they seemed to mostly be holding up. Larry Kenyon devised a clever technique for fixing problems in the ROM by patching the nearest system trap to the problem, with the patch code looking on the stack for the ROM address of the caller, which allowed us to fix problems with tiny, surgical incisions, adding some code snippets here or there, without needing to replace large chunks of the ROM with precious RAM as we had originally envisioned.

By the middle of November 1983, MacPaint was already stable enough to ship, even though it was always skirting on the edge of running out of memory if we changed something in the system. MacWrite still had lots of bugs, but its core functionality seemed to be ready, as long as you weren't pushing memory limits. The only vital application that seemed like it could delay our target date in mid-January was the Finder, which was the shell application responsible for launching other applications and managing files.

Bruce Horn was the only programmer working on the Finder, and he was bogging down with a variety of problems, especially involving file copying in low memory conditions. The Finder was built on top of Bruce's resource manager, using features that were barely finished before the ROM freeze. It probably should have been at least a two man project, but Bruce was a brilliant, passionate, independent perfectionist who insisted that he would get it done in time by himself.

Jerome Coonen, the Macintosh software manager, was afraid that Bruce wasn't going to make it. Clearly, Jerome needed to assign someone else to help Bruce finish, but Bruce's perfectionism and high strung temperament made it hard for someone else to dive in and be helpful, especially this late in the game. But luckily, Steve Capps was the perfect man for the job.

Capps was an extraordinarily talented, creative and prolific programmer, who had transferred over to the Mac team from the Lisa printing group in January 1983, giving us a new burst of energy as the original team members began to tire as our sprint to finish slogged into a marathon. Stockily built, with long brown hair and a bushy beard, Capps was one of the few people around who Bruce respected enough to really listen to, and best of all, he possessed a cheerful, easy-going disposition that was the perfect complement to Bruce's high strung intensity.

To minimize distractions, which were mounting as Apple's marketing machine kicked into high gear as the introduction approached, Jerome arranged for Bruce and Capps to move into a small office a few blocks away, on Bubb Road. Capps dove in quickly and took over a bunch of tasks from Bruce. It wasn't always easy, but by the end of December it looked like the Finder was getting more or less back on track.

Jerome thought of a unique way for the software team to show our appreciation for Capps' heroic effort. Like many hackers, Capps dressed idiosyncratically, almost always wearing a long sleeved, white dress shirt with cut-off denim shorts, white socks, and a distinctive type of checkered sneakers called Vans. In fact, Capps had just given everybody on the team a pair of Vans as Christmas presents. Jerome had the idea that we should pay homage to Capps by declaring "Steve Capps Day", and all come in to work dressed up exactly like him for a day.

I had to go to Macy's to buy a white dress shirt, but I managed to come up with a reasonable fascimile of Capps' attire. It was hilarious to see everyone else as we gathered in our "fishbowl" office in Bandley 3 the next morning, all dressed in Capps' unique sartorial style. Even the French and German translators who were visiting us for the month to get European versions underway joined in. The funniest person was probably Patti Kenyon, who was over 8 months pregnant at the time, wearing an extra large white shirt.

Jerome went to get Capps and Bruce from their enclave, telling them there was an important meeting that required their attendance. We could hardly hold back the laughter as an unsuspecting Capps walked in, and then we finally cracked up and gave him a round of applause when he realized what was going on.

# It Sure Is Great To Get Out Of That Bag!

**Author:** Andy Hertzfeld
**Date:** January 1984
**Characters:** Steve Jobs, Steve Capps, Bruce Horn, Andy Hertzfeld, Susan Kare, Mike Boich, Mark Barton, Steve Hayden
**Topics:** Marketing, The Launch, Software Design
**Summary:** We need a demo for the intro
**Revision:** most recent of 6

It took a monumental effort, fueled by inordinate amounts of chocolate covered espresso beans (see Real Artists Ship) , to finally finish the first release of the Macintosh software in time for the introduction. We finished with literally no time to spare, shipping the "golden master" of the "Write/Paint" disk to the factory at 6am on Monday morning January 16th, a week before the introduction. By that point, most of the software team hadn't slept for days, so we all went home to collapse.

I thought I would need to sleep for at least 24 hours, but I woke up after 6 hours with a desire to go back to Apple to see if the release held up, and to see how everyone else was feeling. By 5pm, most of the software team had dragged themselves back for the same reason, and we were lounging around in a tired daze, happy that we finally shipped, but still not quite believing it, when Steve Jobs strode into the software area.

"Hey, pick yourselves up off the floor, you're not done yet!"

Uh-oh, I thought, someone must have found a show-stopper in the release and we're going to have to track it down. But that's not what he meant.

"We need a demo for the intro! The Mac deserves to have a great demo for its first public showing. I want the Mac to play the theme from "Chariots of Fire" while it's showing a slide show of the apps. Plus lots of other cool stuff, whatever you can come up with. And it needs to be done by the weekend, to be ready for the rehearsals."

We moaned and groaned about being tired, but as we talked we realized that it would be fun to cook up something impressive. We were too tired to think about it right away, but when we came back the next day, a plan started to emerge.

Capps had an idea to use a gigantic font to scroll "Macintosh" across the screen, one letter at a time, to start the demo, so he worked on that, as well as the slideshow. Bruce Horn wanted to do a starry night with twinkling stars, and a skywriter writing "Macintosh" in cursive across the night sky. Susan worked on an intro graphic of the Mac sitting in its canvas carrying bag, and well as some of the other graphics for the slideshow part. I integrated all the pieces and also signed up for the "Chariots of Fire" music part, since no one else wanted to do that.

It's hard to write a music editor/player in two days, but I managed to put something together that could actually play the "Chariots of Fire" theme, but it didn't sound very good, since it used simple sine waves without any envelope shaping. Steve immediately rejected it as lousy, which it was, and opted for using a CD of the "Chariots of Fire" theme to play in the background instead.

Meanwhile, as we were working on the demo, Mike Boich came by with Mark Barton, a third party developer who we seeded with an early Mac, because he had written an impressive program for the Apple II called S.A.M. (the Software Automatic Mouth). SAM was a speech generator, which converted text to speech, with a distinctive, winning personality. I had helped Mark with sound driver issues as he developed it, and now it finally made it to fruition. SAM sounded even better on the Mac, because we had 8 bits per sample and a higher sampling rate.

When Steve heard SAM talk, he immediately decreed that we had to incorporate SAM in the intro demo. "I want the Macintosh to be the first computer to introduce itself!", he insisted. He told Mike Boich to quickly cut a deal with Mark so Apple could bundle the speech generator (rechristened Macintalk) and use it in the intro.

Since my music generator fell through, I got to do the speech part, using Mark Barton's libraries. I knew that I wasn't clever enough to be the Mac's speechwriter. I think Susan had the idea of asking Steve Hayden, Chiat-Day's head writer, to do it. Steve was the guy who conceived the 1984 commercial (see 1984) and was as clever as they come. He was excited about helping out and got it done overnight.

Once we integrated all the pieces together, the demo didn't come close to be able to run on a standard Macintosh. Fortunately, we had a prototype of a 512K Mac in the lab, so we decided to cheat a little (there were only two in existence at the time) and use that for the demo, which made things fit.

The demo starts out with Susan's graphic of the Mac hidden in its carrying bag, on a curtained stage, displayed while the program prepared Capp's big letters. Suddenly, the music swells (from a CD, not generated by the Mac) and Capp's big letters scroll nimbly across the screen, spelling out "Macintosh". Then we transition to Bruce's skywriter, and then to various screen shots of applications, including third party applications like Microsoft's Multiplan and Chart. Finally, the music stops, the screen goes blank, and waits for Steve to press the mouse button. When he does, the Mac starts to speak, in strange but somehow endearing tones:

**Hello, I am Macintosh. It sure is great to get out of that bag!**

**Unaccustomed as I am to public speaking, I'd like to share with you a maxim I thought of the first time I met an IBM mainframe: Never trust a computer that you can't lift!**

**Obviously, I can talk, but right now I'd like to sit back and listen. So it is with considerable pride that I introduce a man who has been like a father to me... Steve Jobs!**

# Can We Keep The Skies Safe?

**Author:** Andy Hertzfeld
**Date:** January 1984
**Characters:** Steve Jobs, Burrell Smith, Regis McKenna, Steve Wozniak, Andy Hertzfeld, Andy Cunningham, Jane Anderson, Katie Cadigan
**Topics:** The Press, The Launch, Marketing
**Summary:** Burrell and I get our pictures in Newsweek
**Revision:** most recent of 18

**Burrell and Andy in Newsweek**

The marketing campaign that launched the original Macintosh was almost as imaginative and innovative as the product itself. It included a carefully orchestrated press blitz, masterminded by Regis McKenna, the legendary Silicon Valley marketing guru whose business card read "Regis McKenna, Himself", and his team of bright, young female assistants, who we nicknamed the "Rejettes": Andy Cunningham, Jane Anderson and Katie Cadigan.

The basic idea was to create a perception of the Macintosh introduction as an epochal event, by garnering as much attention as we could from every possible venue, coordinated to appear around the time of the introduction at the shareholders meeting. Some of the monthly magazines had more than three months lead time, so the press briefings and interviews began in October 1983 with Byte Magazine (see The Mythical Man Year) , and became more numerous with each passing week.

One of the most sought after goals of the press campaign was to obtain a cover story from either Time or Newsweek during the week of the Macintosh launch. Regis and his team were expert at the delicate dance of courtship that such an endeavor required, since journalistic ethics mandated that the cover could not be bought or promised ahead of time. In mid-December, after interviewing Steve and some of the design team, we heard that Newsweek was potentially interested in doing a cover story.

It looked promising enough that the Rejettes arranged an impromptu trip to New York City for Steve Jobs to meet the top brass at Newsweek. Apparently, they were interested in featuring Burrell Smith and myself in the article, too, so we accompanied Steve on a whirlwind three day trip to New York City, staying at Steve's favorite hotel, the Carlyle, which cost over $400 per night. We got a tour of Newsweek's main offices, demoed the Mac to the editorial staff, and even spent some time chatting with Katharine Graham, the long-time publisher of the Washington Post and Newsweek.

After we returned to Cupertino, we got word that we had passed muster and that Newsweek was enthusiastic about doing a cover story about the Macintosh introduction. Burrell and I were interviewed again by Newsweek reporter Michael Rogers, who had written a novel called "Silicon Valley" that I read the previous year, and were scheduled to be photographed the following day.

The software still wasn't finished, even though there were only five days left to work on it, and tension mounted as weeks of relentless pressure were taking their toll on the team (see Real Artists Ship) . When the Newsweek photographer arrived, he wanted to photograph me in my office, but I was afraid that would be too disruptive to the rest of the team, so I told him that I often worked at home, and convinced him to take the picture in the messy office of my house in Palo Alto.

**They used this instead**

Unfortunately, somehow the San Jose Mercury found out that we had been granted the Newsweek cover, and mentioned it in their business gossip column that came out on Friday. Newsweek didn't want the world to think their cover was pre-determined, so that was enough to make them change it - switching, at the last minute, to a stand-by cover asking the burning question "Can We Keep The Skies Safe?"

But it was still amazing to come into work the day before the introduction, and see the January 30, 1984 issue of Newsweek, opened to a four page story about the Macintosh, which began with separate pictures of Burrell and myself in our respective houses. Burrell was sitting on the floor playing his beloved 9-string guitar, with engineering diagrams spread out on the rug in front of him. I was sitting on a chair in the spare bedroom of my house that I used for an office, in its usual state, which was extremely cluttered.

There was a long review of the Macintosh, which was generally positive, although it said that some users may find the graphical user interface to be "visually tiring to use". Burrell had the best quote in the entire article, which was used to end it. When asked what he wanted to do next, he responded "I want to build the computer of the 90's. Only I want to do it tomorrow."

The week following the shareholder's meeting, Steve Jobs and the Mac design team flew to Boston, to recreate the Mac introduction on the East Coast, in front of the Boston Computer Society. Steve Wozniak also came along, and participated in a panel discussion with the design team following the intro spiel.

Burrell and I sat together in a row of seats near the middle of the plane on the six hour flight to Boston. We were just settling in for the long flight when a stewardess approached us, holding a copy of Newsweek, open to the page with our pictures.

"I recognized you from your pictures", she told us, "Can I have your autograph?"

I was flustered, because no one had ever asked me for an autograph before. I demurred, but eventually Burrell and I both signed our names near our pictures, feeling slightly embarrassed.

I turned on my Walkman, pushed back my seat and tried to relax, hoping to be able to fall asleep, but around twenty minutes later, we were approached by a different stewardess, also holding a copy of the Newsweek issue. "Are you the guys who designed the Macintosh?", she asked, "I'd love to get your autographs."

Wait a second, I thought to myself, Woz is on this flight. And sure enough, when I stood up and turned around, Woz was also standing, pointing at us and cracking up. He had convinced the stewardesses to cooperate in pulling one of his typical pranks on us. I started laughing too, because it was a pretty good one.

# Disk Swapper's Elbow

**Author:** Steve Capps
**Date:** January 1984
**Characters:** Steve Capps
**Topics:** Software Design, Technical, Bugs
**Summary:** a last minute bug causes some problems
**Revision:** most recent of 5

One of the more common afflictions of early Macintosh users was the dreaded "Disk Swapper's Elbow" caused by a disk copying operation run amok. Disk swapping was a necessary evil caused by having 400KB floppy disks, 128KB of RAM, and a single floppy drive. If a user wanted to make a backup of a disk, she had to eject the source disk, insert a blank one, format it, drag the source disk over the new disk, and then the Finder would copy data piece-by-piece with the necessary swapping.

A typical application on a 128K Mac had about 85K of memory available; the rest was used by the system, mostly for the bitmap display. A simple calculation shows that copying a 400K disk should involve about 5 or 6 swaps. Five disk swaps was barely tolerable, but, as early Finder users will remember, occasionally it would take well over 20 disk swaps.

You'd start a disk copy and hold your breath during the fifth, and hopefully, final swap. If the Mac dutifully disgorged the floppy that sixth time, you'd convince yourself you miscounted, cross your fingers, and hope for the best. By the seventh swap you start cursing because you knew you were trapped and you'd start wondering about investing in a second, external drive.

Even though the whole Finder was only 46K of code and had about 10K of overhead, the remaining 30K of memory was too small for practical copying. So, I had to break up the code into two chunks: the bare minimum for copying and all the rest. Then, I had to carefully flush out all data that was cached in memory, preload the small disk-copying chunk of code, and coalesce the balance of RAM. Usually, the Finder ended up with 75K of free memory and things worked as planned. But, sometimes the system would mysteriously reload the larger chunk of the Finder code, fragment the free memory, and cause another case of Disk Swapper's Elbow.

It took me a long time to figure out what happened because we had rarely seen this in testing. There were a few bug reports of this problem that were never reproducible. The bug reports went like this: "Copied a disk, it took 20+ swaps $%#@!!! Tried a second time, it was fine." The reason this was not reproducible was because we were all expert mouse users and usually skipped the crucial misstep.

When anybody first starts using a mouse, dragging is one of the more difficult things to do. It's actually quite awkward to click down, move the mouse while holding down the button, and then release. Beginners very often accidentally release the mouse button while dragging. In the Finder, this means you could "drop" an icon you were dragging. You rarely thought about this (unless you happened to drop it over a folder and it disappeared); you'd immediately pick the icon up and continue the drag. It turned out if you dropped the disk icon during a disk copy, you'd induce the bug. Since all of the team members had been using the mouse for years by this time, we rarely dropped icons which is why we could never reproduce the problem.

To support the user's spatial memory, the Finder always remembered where icons were located on the desktop. When you dropped the icon -- even for a half a second -- the Finder would dutifully record its location. The routine to save the icon's location was, as you probably guessed, in the big portion of the Finder's code. When this bug occurred, the Finder would carefully massage the memory for copying and then belatedly discover the icon's location hadn't been flushed out. It would blindly call the routine to flush it and you now know what would happen...

I introduced this bug about 2AM the morning we built the final disks. This bug was caused by a fix to a much more egregious bug, so it was definitely the lesser of two evils....really!

# A Floppy named lsadkfjalhkjh

**Author:** Bruce Horn
**Date:** January 1984
**Characters:** Bruce Horn, Larry Tesler
**Topics:** Bugs, Software Design, The Launch
**Summary:** An early bug in the Finder sends Bruce back to the drawing board...
**Revision:** most recent of 4

After the Mac was introduced in January, 1984, we were all excited to go out to computer dealers and see people's first reactions to our pride and joy. Up to this point, computer users were used to the IBM PC's command-line interface, and we were more than curious as to how people would react to our point-and-click, drag-and-drop graphical user interface. I especially was interested in seeing how people liked using the Finder, and what they found particularly compelling, or difficult.

We went to quite a few stores in the week or so after the introduction, and found that, without exception, every Mac's floppy disk had a garbage name! They were all named something like ";lkakl;rt;klgjh", as if someone had just randomly typed characters to see what would happen. Which is exactly what they did.

In the Finder, the startup disk would appear on the desktop, in the top-right corner, ready to be opened. The Finder would initially select it; once selected, typing would replace the current name, following the modeless interaction model that I had learned in the Smalltalk group from Larry Tesler. This meant that whatever anyone typed when they first came up to the Macintosh would end up renaming the disk.

We had done some user testing, but apparently not enough; our users were already too experienced with the Mac to make this type of mistake. We apparently didn't ask enough people to find out if the Finder passed the "Mom test": would your Mom be able to figure it out without help?

I hurried back to the office and worked on a new release of the Finder that would fix the bug (and others that I had discovered)--it would require the user to take an extra step before she would be allowed to rename the disk.

Finally seeing how the public used the Mac made it clear to me that there were, in fact, good reasons for modes--at least in some cases--and it certainly gave me a taste of humble pie.

# The Times They Are A-Changin'

**Author:** Andy Hertzfeld
**Date:** January 1984
**Characters:** Steve Jobs, John Sculley, Burrell Smith, Andy Hertzfeld, Mike Markkula
**Topics:** Marketing, The Launch
**Summary:** The big day finally arrives
**Revision:** most recent of 23

**Bow-tie era Steve Jobs**

January 24, 1984 - the big day had finally arrived. We had looked forward to the date for so long that it didn't seem real to be actually experiencing the long-awaited public unveiling of the Macintosh at Apple's 1984 annual shareholder's meeting. We were excited, of course, but also nervous about our hastily contrived demo software, and still exhausted from the final push to finish the system software (see Real Artists Ship)

I attended one of the rehearsals held over the weekend, to help set up the demo, and it was fraught with problems. Apple rented a powerful video projector called a LightValve, that projected the Macintosh display larger and brighter than I thought possible. But the Mac had to be connected to the projector through a special board that Burrell cooked up to compensate for the Mac's unique video timings, and the LightValve seemed to be quite tempermental, taking eons to warm up and then sometimes shutting down inexplicably. Plus, Steve wasn't into rehearsing very much, and could barely force himself into doing a single, complete run-through.

Most of the software team usually didn't come to work until after 10am, but this morning we gathered in our fishbowl office in Bandley 3 at 7:30am, so we could walk over together to the big auditorium at Flint Center, which was a half mile away. We got to the cavernous room (which seated up to 2,500) early, but it was already filling up, and soon it was packed tight, with standing room only. The software team sat up close in the second row, in a section reserved for Macintosh division employees.

Finally, the lights dimmed, and Steve Jobs appeared at a podium on the left side of the stage. He was resplendent in a finely tailored black suit complete with a prominent bow tie, looking more like a Las Vegas impresario than a computer industry executive. You could tell that he was nervous as he quieted the rousing applause and began to speak.

"Welcome to Apple's 1984 Annual Shareholders meeting. I'd like to begin by reading part of an old poem by Dylan, that's Bob Dylan", Steve flashed a big smile as he started to recite the second verse of "The Times They Are A-Changin'", stretching an occasional vowel in a Dylanesque fashion:

**Come writers and critics**
**Who prophesize with your pen**
**And keep your eyes wide,**
**The chance won't come again**
**And don't speak too soon**
**For the wheel's still in spin**
**And there's no tellin' who that it's namin'.**
**For the loser now**
**Will be later to win**
**For the times they are a-changin'.**

He thanked Apple's board of directors individually by name for their support in a turbulent year, and then turned the meeting over to Apple's chief counsel, Al Eisenstadt, to run the formal part of the shareholders' meeting. Al ran through some procedural stuff, and then he introduced Apple's CEO, John Sculley, who was just hired nine months ago, for a report on the business.

John reported on Apple's latest quarter, which saw disappointing Lisa sales more than balanced by a fantastic Christmas for the Apple IIe, whose sales had more than doubled from the previous year. But the crowd seemed distracted, impatiently waiting for the main event that was now imminent. John seemed to sense that, and hurried through the bulk of his presentation. Finally he concluded by thanking Mike Markkula and the executive staff for supporting him during his first few months at Apple, thanking one individual in particular.

"The most important thing that has happened to me in the last nine months at Apple has been a chance to develop a friendship with Steve Jobs. Steve is a co-founder of Apple, and a product visionary for this industry, and its my pleasure now to reintroduce Steve Jobs."

Steve reappeared on the left side of the stage as the lights dimmed again. "It is 1958", he began, speaking slowly and dramatically. "IBM passes up a chance to buy a young fledgling company that has invented a new technology called xerography. Two years later, Xerox was born, and IBM has been kicking themselves ever since". The crowd laughs, as Steve pauses.

Steve had cooked up this spiel for the sales meeting in Hawaii last fall, to introduce the 1984 commercial. I had seen him do it a few times by now, but never with as much passion, intensity and emotion, dripping from his voice.

"It is ten years later, the late sixties", he continued, speaking faster now. "Digital Equipment Corporation and others invent the mini-computer. IBM dismisses the mini-computer as too small to do serious computing, and therefore unimportant to their business. DEC grows to be a multi-hundred million dollar company before IBM enters the mini-computer market." Steve pauses again.

"It is now ten years later, the late seventies. In 1977, Apple Computer, a young fledgling company, on the West Coast, introduces the Apple II, the first personal computer as we know it today. IBM dismisses the personal computer as too small to do serious computing, and therefore unimportant to their business," Steve intoned sarcastically, as the crowd applauds.

"The early 1980s. 1981 - Apple II has become the world's most popular computer, and Apple has grown to a 300 million dollar corporation, becoming the fastest growing company in American business history. With over fifty companies vying for a share, IBM enters the personal computer market in November of 1981, with the IBM PC." Steve is speaking very quickly now, picking up momentum.

"1983. Apple and IBM emerge as the industry's strongest competitors, with each selling approximately one billion dollars worth of personal computers in 1983. The shakeout is in full swing. The first major personal computer firm goes bankrupt, with others teetering on the brink. Total industry losses for 1983 overshadow even the combined profits of Apple and IBM."

He slows down, speaking emphatically. "It is now 1984. It appears that IBM wants it all. Apple is perceived to be the only hope to offer IBM a run for its money. Dealers, after initially welcoming IBM with open arms, now fear an IBM dominated and controlled future and are turning back to Apple as the only force who can ensure their future freedom."

Steve pauses even longer, as the crowd's cheering swells. He has them on the edge of their seats. "IBM wants it all, and is aiming its guns at its last obstacle to industry control, Apple. Will Big Blue dominate the entire computer industry? The entire information age? Was George Orwell right?"

The crowd is in a frenzy now, as the already famous 1984 commercial (see 1984) , which was shown for the first and only time during the Superbowl two days ago, fills the screen, featuring a beautiful young woman athlete storming into a meeting of futuristic skinheads, throwing a sledge-hammer at Big Brother, imploding the screen in a burst of apocalyptic light. By the time the commercial is finished, everyone in the auditorium is standing and cheering.

Steve describes the Macintosh as the third industry milestone product, after the Apple II and the IBM PC. "Some of us have been working on Macintosh for more than two years now, and it has turned out insanely great!"

All this time, a lone Macintosh has been sitting in its canvas carrying case near the center of the stage. Steve walks over to the bag and opens it up, unveiling the Mac to the world for the very first time. He pulls it out and plugs it in, inserting a floppy, and the demo begins to run, flawlessly (see It Sure Is Great To Get Out Of That Bag!) . The Macintosh becomes the first computer to introduce itself, speaking in a tremulous voice:

**Hello, I am Macintosh. It sure is great to get out of that bag!**

**Unaccustomed as I am to public speaking, I'd like to share with you a maxim I thought of the first time I met an IBM mainframe: Never trust a computer that you can't lift!**

**Obviously, I can talk, but right now I'd like to sit back and listen. So it is with considerable pride that I introduce a man who has been like a father to me... Steve Jobs!**

Pandemonium reigns as the demo completes. Steve has the biggest smile I've ever seen on his face, obviously holding back tears as he is overwhelmed by the moment. The ovation continues for at least five minutes before he quiets the crowd down.

The rest of the meeting was an anti-climactic blur, as Steve runs through some marketing material and introduces new versions of the Lisa. He shows a slide-show tribute to the Mac team, with voice-overs from the most important contributors. Finally, he turns the meeting back to Al Eisenstadt, to announce the shareholder tallies and complete the formal portion of the shareholders' meeting.

Every member of the audience was given a copy of the first issue of MacWorld magazine, with Steve on the cover, as they departed. Most of the Mac team hung around near the stage, congratulating each other, waiting for the crowd to disperse.

**certificate for gift Mac**

A little bit later, after we returned to Bandley 3, we were surprised by a large Apple truck that pulled up in the parking lot near the back of the building. It contained 100 brand new Macintoshes, one for each member of the team, personalized with a little plaque on the back. Steve presented one at a time to each team member with a handshake and a smile as the rest of us stood around cheering.

We were so keyed up that it was impossible to get back to work that afternoon, but most of us didn't want to go back home, either. The Macs were supposed to go on sale that very day, immediately following the introduction. I thought that it would make it more real to me if I actually could go out and buy one, so five or six of us walked to the nearest Apple dealer, to see if that was possible. The first, closest dealer didn't have any units in stock, and said that they weren't for sale yet, but we didn't give up, and the next dealer was willing to sell me one, even though he didn't have any units in yet either.

# 3rd Party Developers and Macintosh Development

**Author:** David Craig

**Date:** January 1984

**Characters:** Steve Jobs, Andy Hertzfeld, Steve Capps, Bill Atkinson, Bruce Horn, Tom Malloy

**Topics:** 3rd party developers, Lisa, Software Design, Technical

**Summary:** Some Comments about Developing Applications for the Apple Macintosh 128 Computer from a 20 Year Perspective

**Revision:** most recent of 6

Some Comments about Developing Applications for the Apple Macintosh 128 Computer from a 20 Year Perspective

**OVERVIEW**

This commentary contains my recollections about developing 3rd party software for the Apple Macintosh computer during the years 1984 to 1986. This paper was written as my contribution to the Macintosh's 20th anniversary celebrations in February 2004.

During this time I worked for a small company in Wichita Kansas called PPP Inc. (PPP or P3 = Programs for Professional People) which developed a new Macintosh application for the stock market named The Investor. This application was written in Lisa Pascal and contained around 50,000 lines. I and another individual designed the program, I implemented it.

The original Macintosh (called the "Macintosh 128" since it had 128K bytes of memory) provided a fascinating development and application environment which I enjoyed immensely. The Macintosh's rich ROM-based software toolbox in a 64K byte ROM along with the machine's small footprint and superbly clear screen display made the Macintosh a wonderful application platform.

**SOFTWARE DEVELOPMENT USING THE LISA WORKSHOP ENVIRONMENT**

Macintosh development in the early days (circa 1983-1985) was done using the Apple Lisa computer and its Lisa Workshop development environment. I originally used a Lisa 2/5 model which contained 1M byte of RAM, an internal 400K 3.5" Sony floppy drive, and an external 5M byte ProFile hard drive (yes, 5M as in mega bytes was considered a rather large drive in those days). I later used a Lisa 2/10 model which had an additional 10M byte internal Widget hard drive which gave me a total of 15M bytes of hard drive storage.

The Lisa Workshop hosted a command line interface which accessed a wonderful mouse based editor, a Pascal compiler, a 68000 macro assembler, an object file Linker, the RMaker resource compiler utility program, and the MacCom Lisa-to-Macintosh utility communications program.

The Lisa Pascal language was very powerful and compiled Pascal source files to Motorola 68000 object code files. I never found a need to use the Workshop's 68000 assembler since everything I needed for my application could be written in the higher level Lisa Pascal language. Macintosh application resource information was created as text files which were then compiled to a binary format using the RMaker resource compiler. Transferring a Macintosh object program from the Lisa to the Macintosh required the Lisa utility program MacCom which copied Lisa files to a Macintosh formatted disk in the Lisa's 400K internal disk drive. MacCom combined separate Lisa

data and resource fork files which were stored on the Lisa's hard drive and stored them as single documents on the Macintosh floppy.

Macintosh programming was based on a collection of programming libraries called "units" in Pascal parlance. These resided on the Lisa and implemented the Macintosh application programming interface (API) called the Toolbox and Operating System by Apple. These libraries came on Lisa formatted disks called the Lisa Macintosh Supplement. I recall receiving around 3 or 4 supplements each with around a half dozen disks with these libraries. These disks also contained Macintosh utility and sample applications such as the Uriah Heap desk accessory by Andy Hertzfeld (called desk ornaments in the early days), the Edit text editor, and the File application by Cary Clark which showed detailed examples of Macintosh programming.

Macintosh development using a Lisa 1 model was also possible though I never worked with the Lisa 1. For this computer, which did not contain an internal Macintosh compatible 400K byte floppy drive, you transmitted your Macintosh program from the Lisa to the Macintosh via the Lisa's serial using a special Lisa utility. A special Macintosh utility received the transmitted file.

Macintosh development was also done using the Lisa Monitor development environment, but I never used this (I actually did play with the Monitor one time but thought the Workshop was a better environment). The Monitor was the Workshop's predecessor and was also command line based though its command structure was more UCSD p-system based then the later Workshop command structure. I was told that some people at Apple preferred the Monitor to the Workshop since the Monitor compilation and assembly was faster (specifically, Bill Atkinson and MacPaint and the Macintosh Finder team, Bruce Horn and Steve Capps). I've also seen Workshop references in Apple Macintosh source code to the "porkshop" but think this was somewhat unfair.

Around later 1984 or 1985 Apple provided the Macintosh Development System (MDS) which ran on the Macintosh 512K model I recall (I believe the Macintosh 128 couldn't run MDS but may be wrong). This allowed you to develop 68000 assembly language programs on the Macintosh. I never used it since I didn't write in assembly language (too tedious) and I had a Lisa with the Workshop and its wonderful Lisa Pascal language.

Concerning the Macintosh Plus computer which debuted in 1986, this computer was the last Macintosh whose system software was developed by Apple using the Lisa computer and its Workshop environment and the Lisa TLA 68000 assembler. Future Macintoshes were developed using Apple's MPW environment.

Note that the Lisa Workshop also supported a C compiler around 1985, but very little Macintosh development used Lisa C.

**SOFTWARE DEVELOPMENT USING THE MACINTOSH MPW ENVIRONMENT**

I used a beta version of the MPW (Macintosh Programmer's Workshop) programming environment around late 1985 early 1986 for Macintosh development. This was Apple's successor to the Lisa Workshop which was being discontinued since the Lisa hardware had been discontinued in 1985.

MPW ran on the then new Macintosh Plus computer which contained 1M byte of RAM and an internal 800K byte 3.5" floppy drive. I recall using an external floppy drive for my MPW development of The Investor application which worked fine, but compiles were much slower than the Lisa Workshop compiles.

MPW was a very good development environment which I still use today (it now is up to version 3.4 or 3.5 I believe).

## MACINTOSH TOOLBOX AND OPERATING SYSTEM API

Macintosh programming was based on the Macintosh application programming interface (API) called at that time the Macintosh Toolbox and Operating System routines. There were around 500 of these routines in the original Macintosh. As a comparison, I just counted the number of routines in the Macintosh API MPW 3.2 Pascal interfaces from 1990 and there were around 2,300 routines (almost 5 times as many).

The Macintosh API introduced (at least to me) new programming topics such as event based programming, resources, and internationalization of text, numbers, and dates.

One idea that the Macintosh API attempted to teach developers was that the Macintosh was really a software system and not a hardware system. Prior Apple systems (the Lisa excluded) such as the Apple II and III families were more hardware oriented and minimal API information existed. Instead of writing data to a memory location for screen displaying, you instead used the QuickDraw graphics library. Apple wanted Macintosh developers to use the Macintosh API extensively since it already provided most of the core features of applications, ran fast, and was well documented. API usage also tended to promote a standardized user interface which really did not exist for Apple's earlier Apple II and III computers.

The Macintosh Print Manager was a joy to use. It provided a device independent architecture for printing really nice looking text and graphics. The old days of sending printer specific control codes to a printer and hoping for the best were at an end.

The Macintosh Memory Manager and its use of double indirect memory references called handles was an eye opener. This handle architecture provided a simple way to maximize the use of the Macintosh's limited memory size when memory blocks needed resizing; the Macintosh team has to thank Tom Malloy of the LisaWrite word processor team for this (see Hungarian) .

## INSIDE MACINTOSH

The Macintosh API was documented in a wonderful collection of notes called collectively "Inside Macintosh". Originally distributed on a chapter basis these eventually were collected in several volumes. Each chapter documented a specific Macintosh API "manager" such as the Menu Manager. Volumes 1 to 3 from 1984-1985 documented the original Macintosh API information. Volume 4 from 1986 documented the Macintosh Plus and the API changes made for this machine (such as the new SCSI disk manager). Volume 5 from 1988 documented the Macintosh II and its extensive API additions (such as Color QuickDraw).

The early Inside Macintosh chapters also contained API features which were later removed by Apple. For example, the Core Edit manager supported styled text and was a superset of the simpler TextEdit manager. Core Edit was documented in a 1982 or 1983 Inside Macintosh chapter, but was removed from the 1984 Inside Macintosh. Core Edit was used in the original MacWrite word processor.

Inside Macintosh was from my perspective very well written and provided in a very readable fashion a structure which made understanding the Macintosh API much easier. Inside Macintosh's structure was designed from the beginning and all the chapters had the same appearance and readability even though they were written by many different people. Caroline Rose was the key person behind the original Inside Macintosh chapters. She was ably assisted by around a half dozen writers.

Technical notes were also provided as part of the early Inside Macintosh releases. I recall a note from Bill Atkinson describing the internal format of MacPaint documents (he was responsible for the wonderful drawing application MacPaint, the QuickDraw graphics library, and the HyperCard user-oriented "software erector set").

Actual Macintosh system programming sources were also provided as examples. These included all the Macintosh "definition procedures" which implemented features such as window and menu appearances (Andy Hertzfeld wrote these). The sources for the more interesting ROM managers such as the Window or Menu Managers was alas not provided (maybe today, how about it Steve Jobs?).

The User Interface Guidelines chapter was in my opinion the most innovative area in Inside Macintosh. This provided a description of the Macintosh's ideal user interface and a rationale behind the decisions.

Compared to the later book-based Inside Macintosh information that Apple produced around 1990, the original chapter-based Inside Macintosh information was for me more readable and concise. The later material tended to be wordy and overly simplistic.

**LISA MACWORKS**

In 1984 and 1985 Apple supported the Macintosh operating system on the Lisa. This system was called MacWorks and allowed most Macintosh applications to run on a Lisa 2 computer. MacWorks booted the Macintosh OS from a single 400K floppy disk and even displayed the standard "happy Macintosh" boot icon. I recall MacWorks running well as long as the applications you used were well behaved (my Macintosh application The Investor was).

**EARLY MACINTOSH DEVELOPMENT DISAPPOINTMENTS**

Though the original Macintosh provided a revolutionary user interface and application programming interface (API), there were some disappointments from my perspective.

Programming the Macintosh took a long time. Instead of having an application interface consisting of a simple command line interface whose output was a bunch of text lines in a fixed size font, you instead had to manage menus, multiple windows, resources, and events.

Apple could have developed higher level API routines which would have lessened some of the 3rd party development work. For example, in addition to the TextEdit manager, the Macintosh ROM would have contained a TextEdit tool which would have displayed and handled multiple text windows. Unfortunately, this would have required additional programming resources on Apple's part and possibly a larger ROM (say 128K instead of 64K bytes). This type of problem was later solved to some degree by Apple's MacApp object oriented environment but that was many years down the road from 1984.

Sophisticated Macintosh applications required more resources than the Macintosh 128 provided. The original Macintosh's 128K bytes of memory and 400K byte disk drive were on the small size when it came to sophisticated applications (I recall reading that even in Apple there was lots of discussion about this). The original Macintosh was really around a 90K byte memory machine since the screen took 22K bytes of memory and a bit of memory was devoted to system code such a ROM patches and file system buffers. I recall my Investor application was around 200K bytes in size and though it ran on the original Macintosh it was slow due to constant application code segment swapping. The Macintosh 512 ran our program well. It is a shame that the original Macintosh didn't have a bigger memory (I recall reading about a 256K Macintosh) and more disk space (the Macintosh originally used a Lisa 860K byte 5.25" Twiggy floppy drive which would have been wonderful, but in late 1983 Apple changed to the 400K byte 3.5" Sony micro-drive).

From a programming perspective, the Lisa Pascal language was good, but it could have been better. For example, routine and variable names were significant to only 8 characters. This meant that the names such as FlushBuffersNow and FlushBuffersSoon were seen as the same name, FlushBuf, by the Pascal compiler. Apple should have changed the compiler to support at least 16 character name significance, or even better 32 characters. 8 character significance was a real pain for me and reminded me of Apple's Apple II and III Pascal compilers. This naming limitation also caused the Macintosh API routine names to sometimes be very abbreviated.

Macintosh API routine names should have been named to indicate their origins. For example, I thought all Event Manager routine names should have started with EM or EM_ such as EMGetNextEvent or EM_GetNextEvent . This would have at least provided a visual clue in source listing that differentiated your application routines from Macintosh API routines. The Lisa API did this to a far better degree than the Macintosh and both used the same Lisa Pascal compiler.

Macintosh debugging using the MacsBug 68000 debugger was too low-level. I wanted a source level debugger since MacsBug was assembly language based.

Internal Macintosh API data structures should not have been published in Inside Macintosh. Apple knew these were going to change so should not have tempted developers into using this information which can cause incompatible applications when new OS versions are released. This would have been difficult to do given the Lisa Pascal compiler's scoping limitations, but Apple could have changed the compiler to support public and private information better (a Modula-2 reference mechanism could have been useful here from what I know).

The Macintosh API use of global variables was not good (these were also known as "low memory globals"). These promoted the Macintosh as a single process system which later was difficult for Apple to upgrade when it wanted to run real processes on the Macintosh. These global variables also made the Macintosh API non-reentrant which caused problems for interrupt-based tasks.

After using the Lisa and its wonderful Office System during my Macintosh development days I was disappointed that more of the Lisa's software architecture was not implemented on the Macintosh. The Macintosh was based mostly on the Lisa's visual aspects but missed other architectural elements which would have made the Macintosh a better system in my opinion. Too bad Apple could not have better leveraged off of the Lisa's best features to create a Macintosh that was really Lisa version 2 (I know the Macintosh team would cringe at this, but suspect the Lisa team would say that would have been the correct approach which was best for Apple's long term prosperity).

For example, the Macintosh should have supported virtual file names instead of file names tied directly to the file system. The Lisa finder (called the Desktop Manager) supported virtual names containing up to 63 characters even though the low-level file system supported only 31 character names. There could also be multiple Lisa documents with the same name in the same folder. The Macintosh should have also been document-centric and not application centric. Lisa users never dealt with Lisa applications directly (these were called tools in Lisa parlance) but instead always manipulated stationery pads which produced documents.

## MACCOLLEGE

Around the end of 1984 I attended a wonderful Macintosh programming seminar called MacCollege. Held at Apple's Cupertino headquarters it provided a facility with direct access to Lisa computers for development and Apple's original Macintosh technical support team.

Support people such as Scott Knaster, Cary Clark, and Russ Daniels presented Macintosh information, answered programming questions, and helped resolve bugs in your application.

I recall at the end of MacCollege signing a large piece of cardboard paper which had around a hundred names of all the MacCollege graduates (I wonder where this is today?).

## REFERENCES

Here is a list of the key materials that I used during the early Macintosh development days. I still have all these materials including The Investor source code listing and internal architecture manual.

Lisa Workshop manuals (3 volumes, dated 1983 and 1984)

Inside Macintosh manuals (3 volumes for the early days)

MacCollege class notes (around 200 pages)

BYTE magazine and its Macintosh articles (February 1984)

MacWorld magazine premier issue (February 1984)

# Real Artists Ship

**Author:** Andy Hertzfeld
**Date:** January 1984
**Characters:** Andy Hertzfeld, Bill Atkinson, Steve Capps, Jerome Coonen, Bob Belleville, Steve Jobs, Donn Denman, Randy Wigginton, Mike Murray
**Topics:** Management, Software Design, Allnighters
**Summary:** The final push to finish the software
**Revision:** most recent of 7

**Chocolate Covered Espresso Beans**

By the fall of 1983, we had committed to announcing and shipping the Macintosh at Apple's next annual shareholder's meeting, to be held on January 24th, 1984. The failure of the Twiggy disk drive almost caused us to be late (see Quick, Hide In This Closet!) , but it seemed like the new Sony 3.5 inch drive solved all of our problems, and the rest of the hardware was ready to go. The Macintosh ROM was frozen in early September and sent out for fabrication. All that remained was finishing the System Disk, and our two applications, MacWrite and MacPaint.

The software team worked hard over the Christmas break of 1983. The Finder still wasn't finished, and there were lots of performance problems, especially when copying files between disks, which seemed interminable. There was lots of integration testing to do, like cutting and pasting between applications, or applications interacting with desk accessories. As the New Year rolled around, it was clear that we were running out of time.

By the first week of January, the software team was working around the clock, testing and fixing problems that were found. Every employee in the building was drafted as a tester, and we held a series of dinners where Apple bought catered food for anyone who stayed late to test [story:90 Hours A Week And Loving It].

Finally, the deadline for finishing the software was less than a week away, and it seemed obvious that there were still too many bugs for us to ship it. Late on Friday evening, we convinced ourselves that we needed an extra week or two to fix the remaining problems. Steve Jobs was on the East Coast, along with Bob Belleville and Mike Murray, doing press for the introduction, so we arranged for a conference call early Sunday morning to tell him about the slip.

Jerome Coonen, our software manager, spoke for the team, as we gathered around the speakerphone. We were exhausted, and progress was slow. There were still bugs that we hadn't gotten to the bottom of yet, and it didn't seem possible that we could make it in the time remaining. Jerome proposed that we ship "demo" software to the dealers for the introduction, and update all the customers with final software a few weeks later. We thought Jerome was pretty persuasive as we held our breath waiting for Steve to respond.

"No way, there's no way we're slipping!", Steve responded. The room let out a collective gasp. "You guys have been working on this stuff for months now, another couple weeks isn't going to make that much of a difference. You may as well get it over with. Just make it as good as you can. You better get back to work!"

We did manage to wrangle an extra couple of days, by virtue of working the weekend and moving the deadline to 6am Monday morning, when the factory opened, instead of Friday afternoon. We agreed to go home and rest up, and then come into work on Monday ready for the final push.

The final week was one of the most intense I ever experienced. Steve wanted Bill Atkinson and myself to fly to New York to present a Mac to Mick Jagger, but I decided that I needed to stay in Cupertino to help with the bug fixing. Some of us were pausing work to get photographed for magazines like Newsweek and Rolling Stone, which made others on the team feel terrible that they were being left out. At times, the atmosphere got pretty tense.

Friday finally rolled around and it was clear that there were still too many bugs in both the Finder and MacWrite. Randy Wigginton brought in a gigantic bag of chocolate covered espresso beans, which, along with medicinal quantities of caffeinated beverages, helped us forgo sleep entirely for the last couple of days. We starting doing release cycles that were only a few hours apart, re-releasing every time we fixed a significant problem.

When a new release was ready, we would all grab it and start testing again. At one point, around 2am on Sunday night, I stumbled across a bug in the clipboard code. I thought I knew what it might be, but I was so tired that I didn't want to deal with it. I tried to pretend that I didn't see the problem, but Steve Capps was watching my expression and knew there was something wrong. I also was too tired to sustain a pretense; he grilled me about the problem and then helped me craft a fix, since I was too tired to do it on my own.

Around 4am, we had a release where everything seemed to go wrong - even MacPaint was crashing, which was usually rock solid. But our final release, around 5:30am seemed to be much better; the worst problems seemed to have receded and we thought we might actually have a decent release candidate.

We all focused on testing the final release as much as we could until 6am, when Jerome would have to leave to drive it to the factory. It looked pretty good, but soon someone found a potential show stopper - the system seemed to hang when a blank disc was inserted during MacWrite - the disk didn't start formatting like it should. I realized that it was probably hung up waiting for an event, so I reached out and tapped on the space bar, and formatting commenced. Jerome thought the bug was bad enough to hold up the release, but he left to drive it to the factory anyway, figuring they needed to start duplication even if it was just going to be a demo release.

The sun had already risen and the software team finally began to scatter and go home to collapse. We weren't sure if we were finished or not, and it felt really strange to have nothing to do after working for so hard for so long. Instead of going home, Donn Denman and I sat on a couch in the lobby in a daze and watched the accounting and marketing people trickling into work around 7:30am or so. We must have been quite a sight; everybody could tell that we had been there all night (actually, I hadn't been home or showered for three days).

Finally, around 8:30 Steve Jobs arrived, and as soon as he saw us he immediately asked if we had made it. I explained the formatting bug to him, and he thought that it wasn't a show stopper, which meant that we were actually finished. I finally drove home to Palo Alto around 9am and collapsed on my bed, thinking that I'd sleep for the next day or two.

# A Mac For Mick

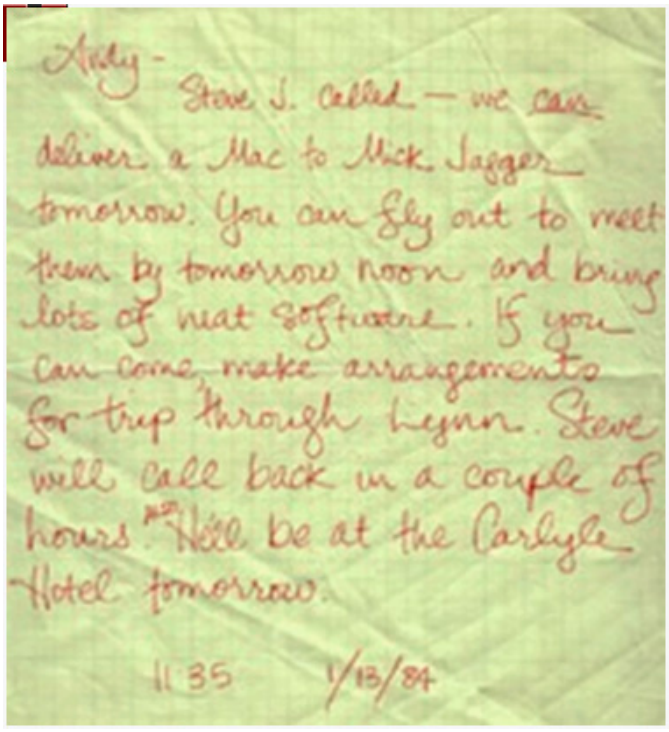
**Author:** Andy Hertzfeld
**Date:** January 1984
**Characters:** Bob Belleville, Steve Jobs, Lynn Takahashi, Patti King, Bill Atkinson, Mike Murray, Andy Hertzfeld
**Topics:** Marketing, Celebrities, The Launch
**Summary:** We present a Mac to Mick Jagger
**Revision:** most recent of 18

**Mick Jagger**

The last couple of weeks before the Macintosh unveiling on January 24th were extremely hectic. The software still wasn't finished, and it wasn't clear if there was enough time left to get it into adequate shape. Meanwhile, the Apple PR machine was revved up to full speed, so there were also plenty of unusual diversions, like being interviewed and photographed for the national press.

The absolute deadline for finishing the software was when the factory opened at 6am Monday morning, on January 16th, eight days before the introduction. When I came into work on Friday, January 13th, I knew that I would probably stay there all weekend, along with the rest of the team, working as hard as possible to shake out the remaining bugs before Monday. Steve Jobs, Mike Murray, Bob Belleville and others were in New York city doing a press tour, so I thought we would be relatively free of distractions, and would be able to focus on bug fixing.

I came into work later than usual, around noon, since I had been at Apple until 3am the previous evening, and I wanted to get one decent night's sleep before the final push. As I went to sit down, I noticed that a handwritten note had been placed on my chair. It was from our software librarian, Patti King, who had taken a message from Steve Jobs' secretary, Lynn Takahashi.


Andy -
Steve J. called — we can deliver a Mac to Mick Jagger tomorrow. You can fly out to meet them by tomorrow noon and bring lots of neat software. If you can come, make arrangements for trip through Lynn. Steve will call back in a couple of hours. He'll be at the Carlyle Hotel tomorrow.
11:35 1/13/84

**actual note**

"Andy - Steve J. called - we can deliver a Mac to Mick Jagger tomorrow. You can fly out to meet them by tomorrow noon and bring lots of neat software. If you can come, make arrangements for the trip through Lynn. Steve will call back in a couple of hours, also, he'll be at the Carlyle Hotel tomorrow."

Wow! A chance to meet Mick Jagger was a once in a lifetime opportunity. But we still had three more days before the deadline, and I would be absent for at least 30 hours if I tried to go to New York, plus I would be relatively useless when I returned from all the flying. I called back Lynn to tell her to tell Steve that I couldn't make it. But I was curious to find out about Mick's reaction.

I found out from Bill Atkinson when he returned from the East Coast on Sunday afternoon, and I got more details from Steve and Mike Murray a bit later. Steve had apparently gone to a party on Thursday evening, where he was introduced to Andy Warhol. Andy got really excited about the Macintosh when Steve demoed it to him. "You must show it to Mick,", he proclaimed, and arranged for Steve and the Apple crew to go to Mick Jagger's townhouse on Saturday afternoon to present him with a Macintosh.

Steve Jobs, Mike Murray and Bill Atkinson got out of the cab in front of Mick's two-story brownstone townhouse, hauling along a Macintosh in its canvas carrying case. They knocked on the door at the address they were given, but there was no response for several minutes. Finally, the door was opened by two huge guys who were obviously bodyguards, who didn't seem all that impressed to be talking to the co-founder of Apple Computer and his entourage.

The Apple folk were led upstairs into an elegantly furnished room to wait for Mick. Bill set up the Mac and launched MacPaint, and started to fool around with it. Then, abruptly, Mick Jagger strode into the room, dressed casually in a T-shirt and blue jeans.

Mick was polite, but he didn't seem to have heard of Apple Computer, Steve Jobs or the Macintosh. Steve tried to strike up a conversation, but he wasn't very successful. Steve told me that Mick couldn't seem to put together a coherent sentence. "His speech was slurred and very slow", Steve described it later, "in fact I think he was on drugs. Either that or he's brain-damaged." After a few minutes, it was clear that Mick had absolutely no interest whatsoever in Apple or the Macintosh, and an awkward silence ensued.

Fortunately, Mick's twelve year old daughter Jade had followed Mick into the room, and her eyes lit up when she saw MacPaint. Bill began to teach her how to use it, and pretty soon she was happily mousing away, fascinated by what she could do with MacPaint. Even though Mick drifted off to another room, the Apple contingent stayed with Jade for another half hour or so, showing off the Macintosh and answering her questions, and ended up leaving the machine with her, since she couldn't seem to part with it.

# Leave Of Absence

**Author:** Andy Hertzfeld
**Date:** March 1984
**Characters:** Andy Hertzfeld, Bob Belleville, Brian Howard, Bud Tribble, Alan Kay, Steve Jobs, Jerome Coonen, Bill Atkinson, Burrell Smith
**Topics:** Quitting, Management, Lisa Rivalry, Personality Clashes
**Summary:** I didn't know how to deal with my bad review
**Revision:** most recent of 24

I didn't know how to deal with the bad performance review that I received from Bob Belleville in February 1983 (see Too Big For My Britches) . Up to that point, I had loved my job at Apple, and was devoting myself to working on the Macintosh, which I passionately believed would change the world significantly for the better. But it was clear that Bob was out to get me, for reasons that I only partially understood.

Jerome Coonen had recently started as the new software manager, so at least I didn't have to interact with Bob directly very often. In fact, Bob seemed to want to avoid me even more than I wanted to avoid him. My initial instinct was to quit, but I believed in the Macintosh too much to leave until it shipped, which was at least six months away, so I resolved to keep working hard while I thought about what I should do.

It seemed like the main problem was that Bob and I had very different views concerning the organization. I worshiped at the altar of the Apple II, and romanticized my work, seeing it more as a calling than a job. I was much more enthusiastic about the computer that we were creating than the engineering organization that was creating it, and I was difficult to manage because I was self-righteous and immature (although I didn't see it that way at the time) and thoroughly disrespected organizational authority.

On the other hand, Bob Belleville saw his job as rescuing the Mac team from the chaotic development process that I thrived in, instilling a modicum of order and predictability, which was necessary to scale the organization. He saw my lack of respect for lines of authority as undermining the organization, which was unacceptable to him. I think that the negative review was intended as a wake-up call, to compel me to change my style to fit his vision of the organization, but he was surprised that I took it as hard as I did.

There didn't seem to be any way to reconcile with Bob, since I never received a written review and he disavowed saying the worst things he told me during our conversation. Besides, I didn't think I wanted to work in the type of organization that he was trying to establish anyway. I decided that I still wanted to work for Apple, but I didn't want to work for Bob, even indirectly. Perhaps the Macintosh team had to eventually mature into a cumbersome organization, but I thought that Apple would still need small teams and people like me to get the ball rolling on something new.

When Bud Tribble left the Mac team to return to medical school at the end of 1981 (see Gobble, Gobble, Gobble) , I considered leaving, too, but Steve Jobs persuaded me to stay, partially by promising to protect me from authoritarian managers. But when I tried to discuss the situation with him over the next few months, he was usually dismissive, belittling the problem and telling me that I didn't have to love Bob to work for him. Sometimes, he would cryptically hint that he had some solution in mind, but nothing ever materialized.

As 1983 drew to a close, I was swept up in the monumental effort to finish the software (see Real Artists Ship) , and then the blissful joyride of the product introduction (see The Times They Are A-Changin')  in January. But by the middle of February, things had calmed down and I knew it was time for me to make a decision about my future at Apple.

My relationship with Bob Belleville had worsened, if that was possible, after he went on a tirade in his staff meeting in December 1983 when he found out that I had assisted Burrell Smith and Brian Howard by writing some diagnostics when they asked for help with the LaserWriter prototype they were working on. Everyone on the software team was exhausted from the high pressure marathon effort to finally complete the software, and tension with Bob made it hard to be enthusiastic about the future.

In February, Apple decided to merge the Macintosh and Lisa groups together, after laying off a quarter of the Lisa people, putting the Mac people in all of the top positions. Steve had always promised us that the group would never exceed 100 people. But now, when combined with more than 200 Lisa folk, it would be over 300 employees strong.

I watched as Steve stood up in front of the assembled Lisa team and announced the merger and layoffs, telling them that they had screwed up and were B or C players. "So, today we are releasing some of your fellow employees to give them the opportunity to work at our sister companies here in the valley," he declared in classic Steve Jobs style.

Someone suggested the alternative of going on a "leave of absence", instead of quitting entirely, which sounded better to me, since I would retain my badge and the prerogatives of an employee when I visited Apple and could more easily return if things seemed better after they settled down. I decided to take a six month leave of absence, starting on March 1st, 1984.

I told Steve Jobs about my leave of absence plans, which he said he regretted, but he didn't offer me any alternative that was acceptable to me. I proposed that we spin off another small team that could work directly for him, now that the division had over 300 people, but he wasn't interested. With the Macintosh finally shipping and the divisions combined, Steve felt he needed managers like Bob Belleville to manage the huge battalion of employees, much more than creative types like myself. Also, he told me that he was sure that I'd be so bored in a month or two that I'd come back from my leave early.

A couple of days before my leave commenced, Steve came into the software area escorting a surprise guest. They came over to my cubicle and Steve introduced me to Apple's newest employee, Alan Kay (see Creative Think) , who had recently departed from Atari and had just signed on as an Apple Fellow. Alan Kay was one of my heros, and it made me even more depressed than I already was to know that leaving would mean that I wouldn't get to work with him.

At the end of my last day of work, the software team held a farewell dinner for me, at a small, fancy continental restaurant called Maddalena's on Emerson street, which was around five blocks from my house in Palo Alto. Now that my last day had actually arrived, I was really sad about leaving all my friends at Apple. I walked over to the restaurant with Burrell Smith, who lived in the house next door to me, wondering if I would be able to survive the dinner without bursting into tears.

Most of the software team came to the dinner, as well as Steve Jobs. I was in a sort of daze as the elaborate dinner was served, followed by some toasts, where people said how much they liked working with me, while I only had sporadic success at holding back tears. Bill Atkinson said that he had no idea of what I would work on next, but he knew that he would be amazed by it. Steve Jobs said that he would miss me, and that he hoped that I would hurry back from my leave. But then he said something strange, apparently commenting on my emotional state: "The thing I like best about Andy is that it's so easy to make him cry".

Finally, the dinner was over and I walked back home with Burrell, still feeling numb, as if I didn't want to think about my conflicted feelings just yet. When I awoke at my usual time the next morning, I had to fight the urge to drive down to Apple as usual. It took a week or two before it stopped feeling strange to not go into work.

# First Day in the Mac Group

**Author:** Scott Knaster

**Date:** April 1984

**Characters:** Debi Coleman, Steve Jobs, Cary Clark

**Topics:** 3rd party developers, Lisa, Documentation, Personality

**Summary:** On my first day, I collect new hardware, get a tour of the building, and witness memorable events.

**Revision:** most recent of 8

In March of 1983, I moved to California to work at Apple. My first job was answering the Lisa Hotline, helping Lisa customers work through their application questions and problems. Although the Lisa was very cool technology in 1983, and I was happy to be working on it, the applications weren't deep, and sadly, there weren't many customers. This left plenty of time to learn about other things, like the Lisa's development system, which was also used to program the Macintosh, due to ship in 1984.

I had a friend, Cary Clark, who worked in the group that helped third-party developers write Macintosh software. Late in 1983, Cary gave me an amazing stack of documentation: an early version of Inside Macintosh (see Inside Macintosh) . It was thrilling to learn how, with simple Pascal procedure calls, any programmer could create windows, track buttons and scroll bars, work with pull-down menus, and draw sophisticated graphics. So when Cary told me there was a job opening in his group, I jumped at the chance to work there.

My first day in the Macintosh group was in April 1984. Cary set me up with the standard set of hardware for developing software: in addition to a Mac (128K, of course), I got a Lisa, used for editing and compiling code, and an Apple III, which acted as a terminal to the Mac for debugging purposes. I also received a most precious accessory: an external floppy disk drive for the Mac. This caseless prototype was such a hot item that I was advised to lock it in a desk drawer at night.

After setting up my hardware, Cary showed me around the Mac building, Bandley 3. We visited the software team's "fishbowl" near the back of the building, the hardware group's space, and the areas for marketing, finance, and other teams. The director of finance, Debi Coleman, often yelled questions loudly to other people across the smallish building, a habit I witnessed on that first day.

The building's atrium featured a couple of stand-up video games, a fancy stereo system with the first CD player I ever saw, and, incongruously, a grand piano and a motorcycle, placed there by Steve Jobs as examples of great product design.

As Cary and I neared the end of our little tour, we came to one of the few closed, non-cubicle offices in the building. It belonged to Steve Jobs. As we walked past, I peeked in and noticed an Apple logo woven into the carpet. Steve was having a conversation with a couple of guys from the Accessory Products Group, the division that made printers, keyboards, modems, and so on.

Steve was concerned about a shortage of ImageWriter dot matrix printers. Apple had forecast that about 70% of Mac buyers would also want an ImageWriter, but the actual figure was over 90%, which led to the shortage. To make things worse, the ImageWriter used a part called a microcontroller that was also in short supply. The printer guys in Steve's office were in charge of a new product called the Wide Carriage ImageWriter, a special edition of the printer that was bought mainly by big accounting firms, and usually with an Apple II, not a Mac.

The Wide Carriage ImageWriter used the same scarce microcontroller, which further reduced the number of regular ImageWriters. Steve was agitated about that, because some customers were refusing to buy a Mac if they couldn't get an ImageWriter at the same time.

As Cary and I walked past Steve's office, we heard him yelling at the printer guys, reminding them that every Wide Carriage ImageWriter built with the hard-to-get microcontroller would likely cost the company a Mac sale. "If you build even one of those Wide ImageWriters…", and then he told them about a certain part of their anatomy that would be "cut off" if that happened. The printer guys looked like they would rather be anywhere else than right where they were. Before too much longer, Apple did ship the Wide Carriage ImageWriter, the microcontroller shortage cleared up, and I always felt privileged to have experienced so much about the Mac division on my first day.

# Spoiled?

**Author:** Andy Hertzfeld
**Date:** April 1984
**Characters:** Bud Tribble, Steve Jobs, Rod Holt, Burrell Smith, Jef Raskin, Randy Wigginton, Hartmut Esslinger
**Topics:** Management, Buildings
**Summary:** The Mac team had a reputation for being spoiled
**Revision:** most recent of 21

**Woz in the Bandley 3 Lobby**

The Macintosh team had a reputation for being spoiled, which was certainly true by the middle of 1984, but it wasn't always the case. Even though Steve Jobs was fond of bragging that the Mac designers were Apple's best engineers, that wasn't necessarily reflected in their compensation.

Two weeks before I transfered to the Mac team after a shake-up in the Apple II group in February 1981 (see Black Wednesday) , I received my regular six month review and was slated for a nominal raise in salary, from $22,000/year to $24,000. I thought that I should still get my raise, even though I had switched groups, so after working on the Mac for a few weeks I approached my new manager Bud Tribble about it.

"Well, that sounds reasonable to me," Bud told me when I explained the situation to him, "but there's a problem. I'm only getting paid $20,000/year."

I was shocked that Bud was making so little, since the average manager in the Apple II group was making at least twice that much. I asked Burrell Smith about his compensation, and found out that he was getting paid even less than Bud, since he started at Apple as a lowly service technician (see It's The Moustache That Matters) , and had hardly gotten any raises as his responsibility grew. Less than two months ago, the Mac was an iffy research project under Jef Raskin, and Steve Jobs hadn't adjusted anyone's salary since he took over.

The next day, I talked with Steve and told him about my incipient raise, and asked him why, if the work we were doing was so important, Burrell and Bud had such low salaries. Steve was unexpectedly nonchalant, professing that he didn't know what their salaries were, and that he hadn't given anyone raises because no one had asked for one. He said that we had much more important things to worry about than our salaries, but he had no problem giving all three of us modest raises right away. But even after the raises came through, the Mac team continued to be relatively underpaid compared to the rest of Apple.

Rod Holt, the designer of the Apple II power supply and the first Macintosh engineering manager, was an extraordinary, opinionated individual who could expound brilliantly on a startling range of topics. He was the unlikely combination of a committed socialist and a multi-millionaire, by virtue of his being one of Apple's earliest employees. On one occasion, the Mac team hired an older

analog engineer to work on the disk controller, and paid him almost twice what Burrell was making, even though he was only doing a small fraction of the work. When we complained, Rod invoked his economic theories, how people should be paid according to their needs instead of their talents. We didn't necessarily buy that, but we really weren't working mainly for the money, and Rod was so charmingly philosophical that we let it slide.

Our offices at Texaco Towers were also kind of quaint by Apple standards. Most of Apple's offices were outfitted with high tech, partitioned Herman Miller cubicles, but Texaco Towers was more old fashioned, with funky, older desks and second hand furniture. Steve was generally tight with money, and usually turned down any extravagant requests. For example, he allowed us to buy an IBM PC to dissect for $2,000 in August 1981 (see Donkey) , but nixed our similar request for a $20,000 Xerox Star.

Burrell and I used to walk down to the nearby Texaco station to get soft drinks from their vending machine most afternoons around 4PM. One afternoon, in the summer of 1981, Steve brought a visiting dignitary by for a demo while we were out, and, frustrated by our absence, decided to have a refrigerator stocked with soft drinks installed so we wouldn't have to miss work time to get beverages, making free sodas our first unusual perquisite.

The team's lifestyle began to change as it grew throughout 1982. In mid-1982, we moved from Texaco Towers to Bandley 4, a more standard Apple building back at the center of the main campus, and the salaries of the early team also rose as we had to pay competitive salaries to newcomers. Bandley 4 was a typical, ordinary Apple building, but it was only intended to be temporary quarters for the Mac team, until the building across the street, the much larger Bandley 3, could be renovated to accommodate us.

The design for our new quarters in Bandley 3, which we moved into in the summer of 1983, showed the first signs of extravagance. The software team was ensconced in a large area with glass doors that we dubbed "the fishbowl", since a passerby could observe us without opening the door. The showpiece of the building was a large atrium in the lobby, with fancy skylights and some interesting furnishings.

To one side of the lobby were two video games, which we had the opportunity to purchase cheaply a month or so after moving in; I paid for Burrell's favorite, Defender (see Make a Mess, Clean it Up!) , while Randy Wigginton contributed Joust. On the other side of the lobby was an expensive stereo system bought by Steve, featuring a compact disc player, which was new technology at the time, and almost 100 CDs, which was just about every one that was released at the time.

Bandley 3 also had a nice little kitchen, near the software area, with a much bigger refrigerator than we had in Texaco Towers. Steve decided that sodas weren't very healthy, and had the refrigerator stocked with expensive Odwalla fruit juices, delivered fresh every day, as well as an assortment of other beverages.

In the spring of 1984, right about the time that I left Apple, the lobby began to fill up with more interesting artifacts, purchased by Steve Jobs on his various travels. There was an outrageously expensive Bosendorfer piano that was soon accompanied by a BMW motorcycle, on display as examples of exquisite craftmanship, although it was rumored that Steve purchased them to impress the industrial designer that he was enamored with at the time, Hartmut Esslinger, whose firm, Frog Design, designed the case of the Apple IIc.

That was right around the time that the 100 person Macintosh Division merged with the 250 person Lisa Division, with the Mac people occupying most of the management roles. The Mac had completed its journey from a funky research project to the center of the company, but I continue to think that it was more fun when we had a lot fewer resources.

# Where'd That Hand Come From?

**Author:** Eric Barnes
**Date:** April 1984
**Topics:** Marketing
**Summary:** How the hand and the mouse came together!
**Revision:** most recent of 3

**Early MacWorld cover, Dec 84 (click to enlarge)**

In the mid '70s, I became San Francisco's first professional male hand model. That's a whole other story in itself. But for three or four years, there were no others in the field and I built a nice (and profitable) reputation, shooting hand shots for all manner of products and companies.

Around April, 1984, I got a call from one of my favorite photographers, Dennis Gray, booking me for an hour, asking me to bring a variety of shirt colors, a couple of sports jackets and a suit jacket.

I showed up, a pale blue shirt was chosen, and we began the shoot. I was simply to hold this weird little plastic thing in a variety of ways. After about 45 minutes, Dennis thought we had the shot. I then asked him, "What *is* this thing?" Dennis replied, "Darned if I know, they call it a 'mouse.'"

A year later, I bought my first Mac, a 512. Unpacking it, I saw that virtually every surface of the boxes, plus all the manuals had this hand on it. Oddly, I didn't remember the shot amidst all the many I'd done in the meantime. It only dawned on me a few days later that it was my hand!

The connection also brought me a variety of hand shots for MacWorld magazine (Dec 84 cover pictured) and I was later told that my hand was the only 'live' thing on MacWorld covers for its first five years of publication.

So though I didn't have anything to do with the creation of the Mac, I did contribute to it's great marketing effort.

The mouse shot, btw, paid me only $100. I don't suppose Steve would be willing to pay residuals at this late date?

# Thunderscan

**Author:** Andy Hertzfeld
**Date:** June 1984
**Characters:** Tom Petrie, Victor Bull, Bill Atkinson, Andy Hertzfeld
**Topics:** 3rd party developers, Hardware Design, Software Design, Technical
**Summary:** A clever device transforms a printer into a scanner
**Revision:** most recent of 27

**brochure for Thunderscan**

The first project that I worked on for Apple after starting in August 1979 was writing low level software for the Silentype printer, a cute, inexpensive thermal printer for the Apple II, that was based on technology licensed from a local company named Trendcom. In typical Apple fashion, we improved on Trendcom's design by replacing their relatively expensive controller board with a much simpler one that relied on the microprocessor in the Apple II to do most of the dirty work.

The only other engineer working on the project was Victor Bull, who was the hardware designer and also the project leader. Vic was smart, taciturn and easy to work with, and I learned a lot from him about how thermal printers worked, as well as how things worked at Apple. We finished the project quickly, and the Silentype shipped in November 1979, less than four months after I began working on it.

In May 1984, during my leave of absence from Apple (see Leave Of Absence) , I received a phone call from Victor Bull, who I hadn't heard from in a couple of years. He had left Apple more than a year ago to work with his friend Tom Petrie at a tiny company based in Orinda named Thunderware, that sold a single product called Thunderclock, an inexpensive calendar/clock card for the Apple II. Victor said that he thought that I might be interested in writing software for an exciting, clever new product that Thunderware was developing for the Macintosh, which he refused to describe over the phone. He invited me to come visit them to check it out.

In early June, I drove up to Thunderware's office in Orinda, which was about an hour's drive from my house in Palo Alto. After I arrived at their modest headquarters, Vic introduced me to his partner, Tom Petrie, and I signed a non-disclosure agreement before they ushered me into a back room to see their demo.

The most popular printer for both the Apple II and the Macintosh was the ImageWriter, a $500 dot-matrix printer capable of rendering bitmapped graphics, that was designed and manufactured by Japanese company named C.Itoh Electronics and marketed by Apple. Virtually every Macintosh owner purchased an ImageWriter, since it was the only printer that was supported by Apple. Tom's demo consisted of an ImageWriter printer hooked up to an Apple II, that at first glance appeared to be busily printing away. But when I looked closer, I noticed that instead of blank paper, there was a glossy photograph of a cat threaded through the printer's platen, and the

printer's black plastic ribbon cartridge was missing, replaced by a makeshift contraption containing an optical sensing device that trailed an umbilical cord back to the Apple II.

Their potential new product, Thunderscan, was a low cost way to temporarily turn an ImageWriter printer into a high resolution scanner, by replacing the ribbon cartridge with an optical sensor and writing some clever software. Since the resolution was determined by the precision of the printer's stepper motors, which had to be very accurate in order to print detailed graphics, Thunderscan, priced at under $200, had better resolution than flat bed scanners costing more than ten times as much. I loved the cleverness of the ingenious concept, and the Woz-like elegance of saving money and adding flexibility by doing everything in software, but there were also a few problems.

The biggest problem was that Thunderscan could only capture one scan line's worth of data on each pass of the print head, which made it nine times slower than regular printing, since the print head could deposit nine dots at a time. This made for frustratingly slow scanning, often taking over an hour to scan a full page at the highest resolution. Thunderscan was never going to win any races.

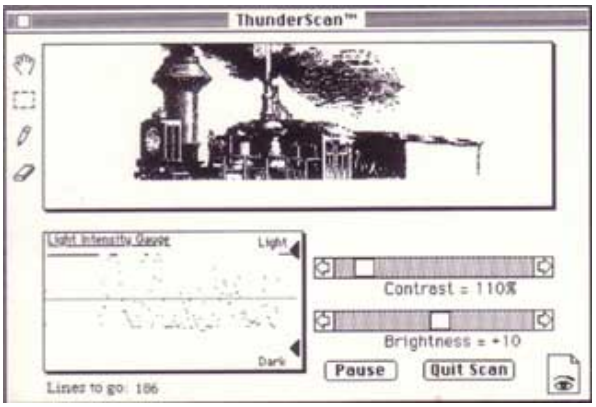
Another apparent problem was the disappointingly low quality of the image being captured and displayed by Tom Petrie's Apple II application. Tom and Vic said their scanner was capable of capturing up to 32 different levels of light intensity, but both the Apple II (in hi-res mode) and the Macintosh only had one bit per pixel to display, so the software had to simulate gray scales using patterns of black and white dots. It looked like Tom was using a simple threshold algorithm to do the rendering, which threw away most of the gray scale information and made the resulting image look unacceptably blotchy. It was hard to tell if the quality promised by Tom and Vic was there or not.

Tom and Vic proposed to hire me to write Macintosh software for Thunderscan. I knew that a low cost scanner would be a great product for the image hungry Macintosh, but only if it had sufficient quality, and I wasn't sure about that. I told them that I'd think it over during the next few days, and, as I did, I became more excited about the potential of Thunderscan for the Macintosh, realizing that the slow speed wouldn't be that much of an impediment if the quality and resolution was good enough. The low image quality in Tom's prototype was probably caused more by the Apple II software than by anything inherent in the scanner. The Macintosh was almost ten times faster than the Apple II, so it should be able to sample the incoming data better to obtain more horizontal resolution. Plus, I knew a much better algorithm for gray scale rendering that would be fun to try out in practice.

My friend and colleague Bill Atkinson was a talented photographer, and one of his hobbies was playing around with digitized pictures, periodically experimenting to find the best algorithms for rendering them. Bill loved to explain his current work to whoever would listen to him, so I learned a lot about rendering gray scale images over the years simply by being around him. Bill had progressed over the years from using an "ordered dither" algorithm, where varying threshold values are specified in a sliding matrix, to his current favorite, which was a modified version of what was known as the "Floyd-Steinberg" algorithm, where an error term is maintained and distributed proportionally to neighboring pixels.

I called Thunderware and told them I was interested in working on Macintosh software for Thunderscan, in exchange for a per-unit royalty. I drove back up to Orinda, where Tom and Vic gave me lots of documentation about the scanner, and the sample code that Tom had written for the Apple II. For the next couple of months, I drove up to Orinda once a week, usually on Thursday, to meet with Tom and Vic show them my progress, prioritize development issues and discussion various complications as they arose. We would also discuss business terms, but we didn't sign a formal contract until the software was almost finished, when we settled on a royalty of $7.50 per unit.

Tom and Vic had already encountered and surmounted a number of tough problems just to get scanning going at all. For example, the ImageWriter printer was not really designed to be stepped one scanline at a time, and if you tried that the paper would bunch up against the platen, causing distortion. Tom and Vic solved the problem by commanding the printer to move three steps up and then two steps back, instead of a single step up, which held the paper snugly against the platen as required. There were also various techniques for sensing the beginning and end of the scan line, and some timings that were determined by tedious experimentation for how long it took the printer to respond to a command.




**The Thunderscan application**

It took a week or so to get basic scanning working on the Macintosh, and then a few more days to render the gray scale data with Bill's modified Floyd-Steinberg dithering. After shaking out a variety of problems, mostly involving synchronization between the printer and the software, I was surprised and impressed by the consistent high quality of the results. I went through a brief, elated phase of scanning every image in sight that would fit through the printer, just to see how it would turn out.

One important design decision that I made early on was to keep the gray scale data around, to allow more flexible image processing. Thunderscan documents were five bits per pixel, before the Macintosh generally supported gray scale, and the user could manipulate the contrast and brightness of selected areas of the image, dodging and burning to reveal detail in the captured image. This also paid off in later versions when we implemented gray scale printing for Postscript printers.

My favorite feature that I came up with for Thunderscan had to do with two dimensional scrolling. Thunderscan documents could be quite large, so you could only show a portion of them in the image area of the window. You could scroll the image by dragging with a MacPaint-style "hand" scrolling tool, but you had to drag an awful lot to get to the extremes of a large image. I decided to add what I called "inertial" scrolling, where you gave the image a push and it kept scrolling at a variable speed in the direction of the push, after the mouse button was released. I had to add some hysteresis to keep the image from moving accidentally, and make various other tweaks, but soon I had it working and it felt great to be able to zip around large images by pushing them.

The hardest feature to perfect was bidirectional scanning. At first, Thunderscan only scanned from left to right, but it wasted time to return the scannner to the left after every scan line. We could almost double the speed if we scanned in both directions, but it was hard to get the adjacent scan lines that were scanned in opposite directions to line up properly. Ultimately, we made bidirectional scanning an optional feature, if you wanted to trade a little quality for greater speed.

I finished the software in November 1984, after taking a short break to work on something else (see Switcher) . Thunderscan shipped in December 1984, and did well from the very beginning, with sales gradually rising from around 1,000 units/month to over 7,500 units/month at its peak in 1987. For a while, it was both the least expensive and highest quality scanning alternative for the Macintosh, although I'm sure it frustrated a lot of users by being too slow. I did three major revisions of the software over the next few years, improving the scan quality and adding features like gray scale printing and eventually gray scale display for the Macintosh II.

Eventually, the flat bed scanners caught up to Thunderscan, and then surpassed it, in both cost, quality and convenience. Over its lifetime, Thunderscan sold approximately 100,000 units and improved countless documents by providing users with an inexpensive way to capture high resolution graphics with their Macintoshes.

# Things Are Better Than Ever

**Author:** Andy Hertzfeld
**Date:** September 1984
**Characters:** Andy Hertzfeld, Bob Belleville, Bud Tribble, Steve Jobs, Pat Sharp, Burrell Smith
**Topics:** Quitting, Personality Clashes, Reality Distortion
**Summary:** My leave of absence was drawing to a close
**Revision:** most recent of 13

**My Apple Badge**

Toward the end of August 1984, my six month leave of absence (see Leave Of Absence) was drawing to a close, and I still hadn't decided whether I would return to Apple. I continued to feel very close to the company, so it wouldn't be easy for me to turn in my badge, but I didn't see a reasonable alternative.

Either way, I was sure that I would continue to write software for the Macintosh, which was still brand new and overflowing with exciting opportunities for innovative applications (see Thunderscan) . I was confident that I could earn more money working independently than Apple was willing to pay me, even if you counted the appreciation of stock options, but financial matters were not my paramount consideration.

The main issue was that I wanted to be able to continue to make a difference in the Mac's evolution and I felt that no matter what I did on my own, it could only have a small fraction of the impact of work done for Apple. Even though things had gone relatively well so far, the Mac's long term success was far from certain, and it was entirely dependent on the moves that Apple made to evolve the platform.

Many of my closest friends were still working on the Mac team, so I heard a lot about what was going on at Apple. I usually drove down to Cupertino to visit them once every week or two, hanging out in the Bandley 3 fishbowl (see Spoiled?) , tentatively at first, but growing more comfortable when I saw that I was still welcome there. I lived next door to Mac hardware designer Burrell Smith, in separate houses on the same lot near downtown Palo Alto, so I heard about Burrell's trials and tribulations at work on a daily basis. Unfortunately, the news wasn't very encouraging.

The Mac team had merged with the Lisa team in Feburary 1984, a few weeks before I started my leave, creating a single large division. At the time, Steve Jobs claimed that the merger would help to transform the rest of Apple to be more like the Mac team, but to me it seemed like the opposite had occurred. The idealistic version of the Macintosh team that I yearned for had apparently vanished, subsumed by a large organization of the type that we used to make fun of, riven with bureaucratic obstacles and petty turf wars.

The core software group was still recovering from the intense effort to ship (see Real Artists Ship) and hadn't done very much all spring and summer, suffering from a classic case of massive post-partum depression. The LaserWriter printer was the current main focus of development, along with the AppleTalk network required to support it, and the core software team didn't have much to do with either. No one had set a compelling new goal for the team, and now it was just drifting.

Burrell Smith had completed the LaserWriter digital board and moved on to work on the "Turbo Macintosh", a new Macintosh digital board featuring a custom chip that supported 4-bit/pixel gray scale graphics and a fast DMA channel to interface an internal hard drive. But Burrell frequently complained of sparring with engineering manager Bob Belleville and others on his staff over trivial design decisions. He thought that Bob didn't really want to add a hard drive to the Mac, favoring the development of a Xerox style "file server" instead, and was therefore trying to surreptitiously kill the Turbo project. I didn't think that Burrell would put up with it much longer; as he phrased it, he was "asymptotically approaching liberation" from Apple.

The one saving grace was that Bud Tribble had finally completed his six year M.D./Ph.D. program at the University of Washington and decided to forgo practicing medicine in favor of returning to his old job at Apple as Macintosh software manager, working for Bob Belleville. In July 1984, he moved into a spare bedroom at Burrell's house in Palo Alto, next door to mine, so I got to see him frequently. I still had the highest respect for Bud, and I loved to show him whatever I was working on because he always managed to improve it with an insightful suggestion or two.

I had mixed feelings about returning to the lumbering Macintosh division, but Bud was a strong link to the good old days and I thought that perhaps we could establish a little outpost in the large organization where the original Macintosh values could prevail. But that didn't seem possible if Bud worked for Bob Belleville, my nemesis whom I blamed for many of the problems. The only solution I could think of was for Bud to work directly for Steve Jobs instead of working for Bob. Bud was all for it, but only Steve could make it happen. I called Steve's secretary Pat Sharp, and arranged to have dinner with Steve and Bud to discuss my possible return to Apple.

We met in the lobby of Bandley 3 and walked to an Italian restaurant on De Anza Boulevard a few blocks away. Steve seemed a bit preoccupied, and I was nervous about how he would react to what I had to say, because I had to implicitly criticize him to make my case. After we ordered dinner I cleared my throat and tentatively plunged ahead.

"As you know, I care a lot about Apple, and I really want to return from my leave of absence. I'd love to work for Bud again, but things seem really messed up right now." I paused for a moment as I gathered my resolve. "The software team is completely demoralized, and has hardly done a thing for months, and Burrell is so frustrated that he won't last to the end of the year..."

Steve cut me off abruptly with a withering stare. "You don't know what you're talking about!", he interrupted, seeming more amused than angry. "Things are better than ever. The Macintosh team is doing great, and I'm having the best time of my life right now. You're just completely out of touch."

I couldn't believe what I was hearing, or tell if Steve was serious or not. I looked to Bud, who communicated his bewilderment with an apologetic shrug of his shoulders, but I could see that he wasn't going to corroborate my views.

"If you really believe that, I don't think there's any way that I can come back," I replied, my hopes for returning sinking fast. "The Mac team that I want to come back to doesn't even exist anymore."

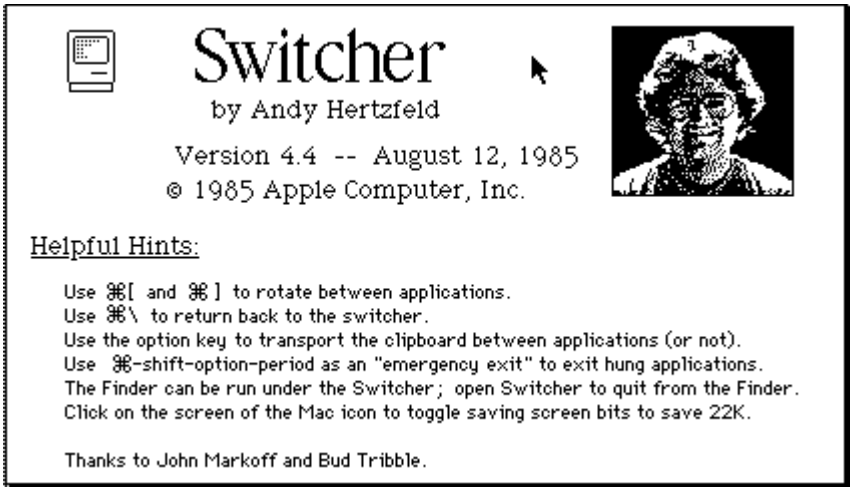
"The Mac team had to grow up, and so do you," Steve shot back. "I want you to come back, but if you don't want to, that's up to you. You don't matter as much as you think you do, anyway."

I saw that we were so far apart that there was little point in continuing the conversation. We finished dinner quickly and walked back to Apple without further discussion.

Actually, quitting was easier than I thought it would be; I just called up Apple's HR department and let them know that I wouldn't be coming back. I didn't even have to sign any paperwork or turn in my badge, which I still have today, almost twenty years later. I had thought it would feel devastating to finally resign, but instead I actually felt relieved for the situation to be resolved, and optimistic about writing Macintosh software on my own.

# Switcher

**Author:** Andy Hertzfeld
**Date:** October 1984
**Characters:** Andy Hertzfeld, Neil Konzen, Bill Gates, Bud Tribble, Steve Jobs, Steve Capps, Charles Simonyi, Phil Goldman, Tom Petrie, Victor Bull, John Markoff, Burrell Smith, Guy Kawasaki
**Topics:** Software Design, Technical, Compensation
**Summary:** The Macintosh's first multi-tasking environment
**Revision:** most recent of 48




**The Switcher About Box**

The first commercial product that I worked on after going on leave of absence from Apple in March 1984 (see Leave Of Absence) was a low cost, high resolution scanner for the Macintosh called Thunderscan, that I created in collaboration with a tiny company named Thunderware (see Thunderscan) . I started working on it in June 1984, and by early October, it was almost complete.

Tom Petrie, one of the two principals at Thunderware (the other was Victor Bull, who I worked with on the Silentype thermal printer, the first project that I did for Apple), arranged a few demos for various computer magazines, to curry favorable reviews to promote the product. On October 11th, 1984, I drove with Tom to an office in Hillsborough, to demonstrate Thunderscan for Byte magazine.

The Byte reviewer was John Markoff, a technology scribe for the San Francisco Chronicle who was one of the best reporters covering the personal computer industry. Tom described Thunderscan while I set up the demo and started scanning. John asked a few questions, taking notes with his IBM PC, which was running a character-based text editor that I viewed with the typical pious disdain of a Macintosh purist. As I was answering one of his questions, the phone rang.

"Excuse me," he told us, as he pressed a key combination on his keyboard, his monitor screen instantly changing to a different program. He talked on the phone for a minute or two, occasionally typing, before he finished the conversation and pressed a key combination to switch back to his Thunderscan notes.

"What did you just do?," I asked John, curious about the software that he was running. "How did you switch to another application so quickly?"

"Oh, I'm running Memory Shift. Haven't you seen it?" John responded. "It's a DOS utility program that keeps multiple applications resident in memory, and allows you to switch between them quickly. I've been using it a lot lately." John typed the switch command a few times in rapid succession, to show me how fast it could do its thing.

"You know, I think I could do that for the Macintosh", I suddenly blurted out, before I even thought about it consciously.

The 512K Macintosh, with four times the memory of the original, had just started shipping a few weeks ago. I had considered trying to run multiple applications simultaneously on the 512K Mac, but I was stymied by low memory conflicts and other potential gotchas. But now, as I observed Memory Shift in action on John's PC, I suddenly saw a simple way to do it, which didn't seem to be too hard to implement.

"Yeah, that would be cool," John agreed. I continued with the Thunderscan demo, but it was hard for me to concentrate on it, because I couldn't stop thinking about application switching. There were a few intricate problems to solve, but it seemed eminently doable, and I thought it would be incredibly useful if I got it to work.

Tom Petrie noticed how excited I was about the new idea and, as he drove me home from the demo, he reminded me of my prior commitments, and made me promise that I would finish the alpha release of Thunderscan as soon as possible, before daring to start something new. We agreed on a list of a dozen or so tasks, which I thought I could accomplish within two weeks. Once the alpha release was completed, I could take a short hiatus, in order to work on application switching, before diving back into Thunderscan to finish the 1.0 release, due by the end of November.

During the next two weeks, I focused on polishing Thunderscan to get it ready for the alpha release, but I also spent idle moments pondering the design of the application switcher. One fundamental decision was whether or not to load all of the applications into a single heap, which would make optimal use of memory by minimizing fragmentation, or to allocate separate "heap zones" for each application. I decided to opt for separate heap zones to better isolate the applications, but I wasn't sure that was right.

There were lots of little problems to solve. The most crucial one was that the system software kept lots of application-specific global variables in low memory (see Mea Culpa) , which needed to be swapped during context switching, so each application could maintain its own set of them. The hard part was coming up with the precise list of exactly what needed to be swapped; many of the variables were obvious, but some were quite subtle, and dependent on how applications were using them. I knew that my first cut wouldn't be perfect, but I was confident that I could debug the inevitable problems once I saw how the applications were actually failing.

A few days after starting the push to finish Thunderscan, I received an intriguing phone call from Jeff Harbers, the manager of Microsoft's Macintosh applications team. Jeff told me that Microsoft had a very strategic project that they needed for the Macintosh, and that they thought that I was the ideal person to implement it. He wouldn't tell me anything else over the phone, but he offered to fly me up to Seattle to discuss it in person. Even though I was right in the middle of trying to complete Thunderscan, I was intrigued enough to accept his offer to fly up to Seattle and visit with him the following Tuesday.

Jeff picked me up at the airport, and we drove to Microsoft's main building where we were joined by Neil Konzen, a talented 23 year old who was Microsoft's main systems programmer on the Macintosh. I knew Neil from his days as an early Apple II hobbyist, when we collaborated on adding features to an assembly language development system when he was only 16.

Jeff asked me what I was working on, and I told him about Thunderscan, which he seemed to be interested in. But when I mentioned that I was about to start some experiments with an application switching utility, his jaw dropped, and he looked like he couldn't believe what I said.

"That's just what we wanted to talk with you about!", he exclaimed, "It's great that you're already working on it".

Jeff explained that Microsoft had put a lot of effort into getting their applications to run well in the tiny space available in the 128K Macintosh, which they considered to be a key competitive advantage. But as things stood, the 512K Mac would undermine their efforts, since it allowed applications to be much larger. Plus, Lotus had recently announced an integrated application suite for the 512K Macintosh called Jazz that made it easy to quickly switch between different functional areas. But if the Macintosh could run multiple applications simultaneously, the small memory footprint of the Microsoft apps would continue to be advantageous, since their lower memory requirements meant that more of them could run concurrently, and users could put together customized application suites on their own. The purpose of the visit was to convince me to write an applications switcher under contract to Microsoft.

Neil Konzen had contemplated a potential design, which he conveyed to me in front of a whiteboard. He decided to use the single heap approach that I had rejected, along with a few interesting twists to minimize memory fragmentation. I told him about the alternate approach of using separate heapzones, and how I thought that it was probably worth it to trade some memory fragmentation for greater robustness. I told him that I would give his approach some more thought.

Finally, my afternoon at Microsoft culminated in a private meeting with their CEO, Bill Gates. Jeff ushered me into Bill's office and reviewed the afternoon's discussions for him, before excusing himself to leave us alone to negotiate a development deal. I had met Bill a few times during the course of Macintosh development, and while I respected his understanding of technology, I was wary of his burgeoning reputation as a conniving businessman.

After exchanging a few pleasantries, and telling me how much the Macintosh mattered to Microsoft, he looked me in the eye and said, "You're a really good programmer, right? I think you must be a really good programmer."

"I guess so," I responded, not understanding why he was attempting to flatter me.

"Well, I think you are. How long do you think it will take to do this project? A month or two? I think a really good programmer like you could get it done in less than two months."

"I really have no idea," I replied, "I'm not far enough along to know if it's even feasible yet."

"Well, let's figure it out," he said in a slightly condescending tone. "I don't think it could be more than 10,000 lines of code, and a really good programmer like you should be able to write at least a thousand lines of code per week, so I think it will take you less than 10 weeks to write it, if you're as good as I think you are."

I didn't know how to respond, so I kept quiet and let him continue.

"And how much do you think a really good programmer should get paid? Around here we pay our best programmers around two thousand dollars per week. Do you think you should be paid more than that?"

"I don't know," I replied. I was finally beginning to see where he was coming from. Bill was trying to get me to brag that I could write the application switcher really quickly, so he could justify paying me a lower price for it.

"Well, I don't think that you could expect to get more than four thousand dollars per week, tops. Actually, I think that's too much, but let's go with that. If it takes ten weeks, and you get paid four thousand dollars per week, that means you should get paid $40,000 for writing it."

$40,000 didn't sound like very much to me, especially if it was as strategic to Microsoft as it seemed to be. I think Bill was expecting me to make a counteroffer, but I wasn't very enthusiastic about selling it to Microsoft regardless of compensation, since it really should eventually be part of the Mac OS.

"Listen, I really want to write this, completely independently from you guys, so you won't have to pay me anything to do it. I certainly don't want to negotiate a deal until I see how it turns out; there might be a show-stopper and I won't be able to get it to work at all. And if I pull it off, it really should be bundled with every 512K Macintosh."

Bill shifted his tactics. "OK, I don't really care if Microsoft owns it as long as it's available to our users. I want you to commit that you'll apply your best efforts to making sure that it runs well with our applications, and that you'll call Jeff if you run into any snags. We can talk again about publishing it later if you want to after you're further along. How does that sound?"

I told him that sounded good, and I promised to do my best to make it work great with Microsoft's applications, which I wanted to do anyway, since the Microsoft apps were important to most users. We shook hands and I departed on a positive note.

By the time I returned to my house in Palo Alto that evening, I was burning with the desire to see if I could get something going quickly. Even though still I had a few more days of work to complete the Thunderscan alpha release, I decided to see if I could write a proof of concept prototype of the application switcher first.

I would eventually have to write a user interface for selecting applications, but the proof of concept didn't have to worry about that; it was hardwired to run MacPaint, MacWrite, MacDraw and the Finder. I worked for 20 hours straight writing the core of the program, which worked by patching traps to extend a few essential system calls, like GetNextEvent, Launch and ExitToShell. The hardest part was going through all of the low memory locations, determining what needed to be swapped. It was incredibly satisfying to see it begin to work, crashing all the time at first, but gradually stabilizing as I tracked down various problems.

I had it working for an hour or so when I saw Bud Tribble return home. Bud had finally finished his M.D./Ph.D. program at the University of Washington, including interning for a year, but he decided that working on the Macintosh was more fun than being a doctor, so he had returned to Apple in his old job as software manager a few months earlier, in July 1984. Bud was living at Burrell Smith's house, which was next door to mine.

I dragged Bud over to my house to show him how I could rapidly switch between MacPaint, MacWrite and MacDraw. He was impressed, but to my surprise he complained that the switching, which was almost instantaneous, was actually too fast.

"I think it might be confusing to switch from one application to another without any feedback," he told me. "What if someone switches accidentally? Maybe you could use animation to make a smoother transition."

That sounded like a great idea to me. We decided that one application should scroll off the screen horizontally while another was scrolling on, which gave the users a simple, concrete mental model of the applications wrapped around a sort of Lazy Susan, which they could rotate to move the desired application to the visible area. I quickly wrote some fast scrolling routines, and was blown away by how cool it looked to see the applications zip across the screen.

I started showing my proof of concept demo to my friends at Apple, as well as a few user group demos, and it was very positively received. Unfortunately, I still had to finish up the product release of Thunderscan, which I barely managed to do by the end of November as I had promised. After taking a short vacation, I got back to work on what I was now calling "Switcher" in early December and by Christmas I had much of it implemented, including a simple UI for selecting applications and "Switcher Documents" for remembering sets of related applications.

In early January 1985, I got a phone call from Guy Kawasaki, one of Apple's third party evangelists, who told me that Apple was interested in buying Switcher, and that he was assigned to make that happen. The first step was arranging a demo for Steve Jobs.

I entered Steve's office with a bit of trepidation, because I thought that Switcher was worth at least a quarter of a million dollars to Apple, but I was sure that Steve would never want to pay me that much. But I was also proud of Switcher, and was interested in seeing how Steve would react to it.

I booted up my by now standard demo of MacWrite, MacPaint, MacDraw and the Finder, as well as a little maze generating program written by Steve Capps. I configured Switcher with the scrolling animation initially turned off, so it would have more impact when I showed it later. I demoed cutting and pasting between MacWrite, MacPaint and MacDraw, in seconds instead of minutes, and then I turned on the scrolling animation, and starting switching rapidly between them, in both directions.

"OK, I've seen enough, " Steve interrupted me. "It's great. Apple is going to bundle it with the Mac. Congratulations."

But then he paused, and stared at me for a moment with an incredibly intense gaze, as if he was sizing me up or maybe just trying to scare me.

"But I don't want you taking advantage of this situation. I'm not going to allow you to take advantage of Apple."

"What do you mean?" I asked him, genuinely puzzled.

"There's no way that you could have written that program without confidential information that you learned by working at Apple. You don't have the right to charge whatever you like for it."

I started to get angry. "The program is only half finished, and if I don't think you're paying me fairly, I won't be motivated to finish it."

Steve gave me another intense stare as he paused for a few seconds. Then he stated a single number, without explanation.

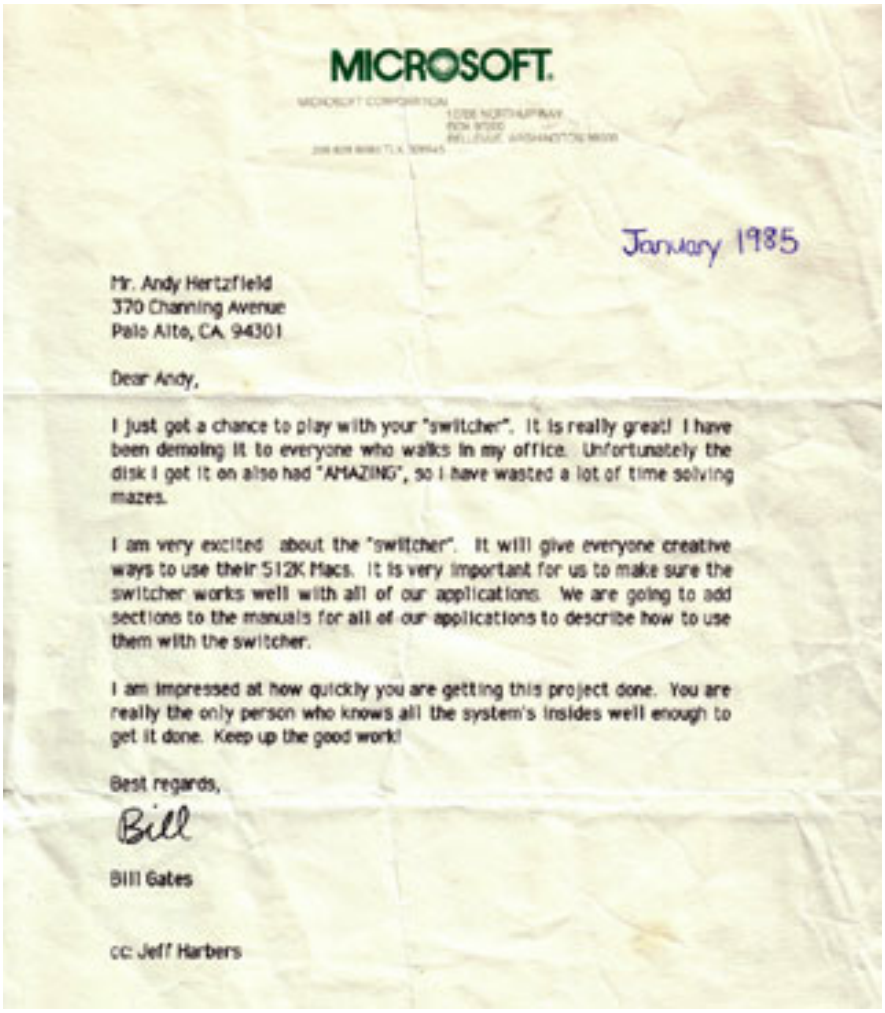
"One hundred thousand dollars."

"I don't know," I told him, "I think it's probably worth a lot more than that."

"Don't argue with me. $100,000 is fair, and you know it."

I didn't seem to have any alternative but to capitulate to Steve's price setting, since he's difficult to argue with and I really wanted Switcher bundled with the Mac. I eventually negotiated the final agreement with Guy Kawasaki, where, in addition to the $100,000, I managed to get a 10% royalty of the wholesale price if Apple sold Switcher separately, which Steve swore they would never do, but eventually the royalty delivered another $50,000.

Getting Switcher going wasn't that hard, but it was a very difficult program to finish, because it overturned some of the underlying assumptions that applications were making, yet it was committed to keeping everything working anyway. Much of the work in the latter stages involved testing it with every application that I could get my hands on and debugging crashes. Usually I could concoct some kind of technique that would mitigate the problem without causing worse problems elsewhere.



MICROSOFT

January 1985

Mr. Andy Hertzfield
370 Channing Avenue
Palo Alto, CA 94301

Dear Andy,

I just got a chance to play with your "switcher". It is really great! I have been demoing it to everyone who walks in my office. Unfortunately the disk I got it on also had "AMAZING", so I have wasted a lot of time solving mazes.

I am very excited about the "switcher". It will give everyone creative ways to use their 512K Macs. It is very important for us to make sure the switcher works well with all of our applications. We are going to add sections to the manuals for all of our applications to describe how to use them with the switcher.

I am impressed at how quickly you are getting this project done. You are really the only person who knows all the system's insides well enough to get it done. Keep up the good work!

Best regards,

Bill

Bill Gates

cc: Jeff Harbers

**Thank you note from Bill Gates**

Predictably, the hardest part of finishing Switcher was making it work smoothly with the Microsoft applications. That was partially because Microsoft was a very early developer, and took liberties with the system that most developers would shy away from, but it was mostly because the instructions comprising their applications were encoded in pseudo-code to save space, in the tradition of the byte-code interpreters from Xerox, which Charles Simonyi advocated.

Unfortunately, the pseudo-code kept me from disassembling the program when it crashed, which made it more difficult to debug. I finally developed a debugging technique by single stepping through their interpreter, six instructions at a time to get to the instructions that were doing the work, but it was pretty painful. I was determined to slog through it because of the promise that I made when I visited Microsoft.

One of the last problems that I addressed before finishing the first release of Switcher in March 1985 had to do with applications hanging. If you're running multiple applications, you don't want one application hanging to take down all of them. I added a feature that allowed the user to kill the current application if it was hung up, by monitoring for a specific key combination during the vertical blanking interrupt handler.

I knew that I had to pick a very rare key combination, because you didn't want users killing their applications accidentally. I decided on shift-command-option-period, four keys held down at once, which I thought would be pretty hard to stumble into accidentally. But I was surprised when I got a call from Jeff Harbers at Microsoft.

"Hey, I like that abort feature that you just added, but you're going to have to change the key combination, because we're using that in Microsoft Word.", Jeff told me. Microsoft Word was very complex, and it possessed an enormous range of keyboard shortcuts, way too many, as far as I was concerned.

"OK, suggest something for me to change it to and I'll consider it," I told Jeff.

Jeff didn't have anything specific in mind, so he told me that he would get back to me soon. I had to laugh when he called me back the next day, and told me that he wanted to withdraw his request and that I should keep shift-command-option-period as the abort sequence.

"OK, that sounds good to me, " I told him. "But why the change? Doesn't it still conflict with Word?"

"We'll change Word in the next release not to use it. The problem was that we couldn't find a safe sequence - I guess we're already using every key combination!"

I officially released the finished version of Switcher in April 1985, and I maintained it for a few versions after that. Eventually, I handed it off to Phil Goldman, a tremendous young programmer from Princeton who was recently hired at Apple, who went on to write MultiFinder (with Erich Ringewald), Switcher's eventual successor, in 1987.

# Handicapped

**Author:** Andy Hertzfeld
**Date:** 1985
**Characters:** Steve Jobs, Jean-Louis Gassee, Andy Hertzfeld, Steve Wozniak
**Topics:** Personality Clashes, Personality
**Summary:** The reason that Steve parks in the handicapped space
**Revision:** most recent of 4

*Most of the anecdotes that I've written for Folklore are based on incidents that I observed myself, but sometimes a second or third hand story is just too good to pass up. I have to issue a disclaimer here that I didn't actually witness the punch line to this one, and it certainly seems too good to be true.*

Steve Jobs was not the most considerate individual at Apple, and he had lots of ways to demonstrate that. One of the most obvious was his habit of parking in the handicapped spot of the parking lot - he seemed to think that the blue wheelchair symbol meant that the spot was reserved for the chairman.

Whenever you saw a big Mercedes parked in a handicapped space, you could be sure that it was Steve's car (actually, it was hard to be sure otherwise, since he also had a habit of removing his license plates). This sometimes caused him trouble, since unknown parties would occasionally retaliate by scratching the car with their keys.

Anyway, the story is that one day Apple executive Jean-Louis Gassee, who had recently transferred to Cupertino from Paris, had just parked his car and was walking toward the entrance of the main office at Apple when Steve buzzed by him in his silver Mercedes and pulled into the handicapped space near the front of the building.

As Steve walked brusquely past him, Jean-Louis was heard to declare, to no one in particular - "Oh, I never realized that those spaces were for the emotionally handicapped...".

One day in October 1983 I got a phone call at my desk at Apple from the Cupertino police department saying something like, "You reported that Mercedes parked in the handicapped space at your lot at Apple. Well, we sent a car out there but we can't really tow it away because the handicapped space is improperly designated."

I had no idea what he was talking about. A few hours later, I found out that Apple's other cofounder, Steve Wozniak, who was a prolific prankster, called up the Cupertino police and reported that a silver Mercedes was illegally parked in a handicapped space and told them the person reporting it was Andy Hertzfeld, giving them my phone number at work. I decided not to inform Apple's facilities department about the improperly marked space, just in case Woz decided to try it again.

# Are You Gonna Do It?

**Author:** Andy Hertzfeld
**Date:** February 1985
**Characters:** Burrell Smith, Steve Jobs
**Topics:** Management, Quitting
**Summary:** Burrell quits Apple, but fails to live up to his threat
**Revision:** most recent of 1

No matter how much resolve you could muster, it was still difficult to quit Apple if Steve wanted you to stay. You'd have to sit down with him for a reality distortion session, which was often effective at getting people to change their minds. One day, a few of us were talking about strategies to overcome Steve's persuasiveness.

"I've got it!", said Burrell. "I know the perfect way to quit that will nullify the reality distortion field."

Of course we wanted to know how he could do that.

"I'll just walk into Steve's office, pull down my pants, and urinate on his desk. What could he say to that? It's guaranteed to work." We laughed, thinking that not even Burrell would have the guts to do that.

A year and a half later, it finally was time for Burrell to quit, after months of scuffling with Bob Belleville and some of the other managers, who wanted to cancel the "Turbo Mac" project (a redesigned, faster Mac based on a semi-custom chip, with an internal hard drive) that Burrell was working on with Brian Howard and Bob Bailey.

Burrell told Bob Belleville (who was probably relieved, since he knew that Burrell didn't respect him) and the human resources department that he was quitting, and then made an appointment to see Steve that afternoon. When he walked into Steve's office, he was surprised to see Steve grinning at him.

"Are you gonna do it? Are you really gonna do it?", asked Steve. Somehow, word about the urination threat had gotten back to Steve, and he was genuinely curious if Burrell would go through with it or not.

Burrell looked Steve in the eye. "Do I have to? I'll do it if I have to."

Steve's expression gave him the answer, so Burrell turned and walked out of the office, no longer an Apple employee.

# Font Manager Redux

**Author:** Andy Hertzfeld

**Date:** March 1985

**Characters:** Bud Tribble, Steve Jobs, Jerome Coonen, Bill Atkinson, Bruce Horn, Larry Kenyon, Burrell Smith, Steve Capps

**Topics:** Software Design, Technical, Personality, Allnighters, QuickDraw

**Summary:** I continued to work on the Font Manager even after I left Apple

**Revision:** most recent of 15

Innovation often requires discarding finished work when a better solution comes along. Each new improvement may impact prior work, so you have to be willing to retool the older parts of a design to better integrate the newer parts as they emerge. One example of this for the Macintosh was the development of font manager, which had to be rewritten a few different times as the system evolved.

Bill Atkinson's QuickDraw graphics package did all the work of measuring and drawing text, but it didn't want to deal with system-dependent details like reading from the file system. The font manager's main job was to load fonts for QuickDraw, given a font family, a size and a style, so Quickdraw could just blast out characters without having to worry about how fonts are stored and loaded.

The initial implementation of the font manager simply included a few built-in fonts that were linked with the system, and it returned the system font if you requested one that wasn't built-in. The initial system font that we used through most of 1981 was one that we borrowed from Smalltalk called "Cream".

In the spring of 1982, the first implementation of our user interface software was beginning to come together. We wanted to allow application writers to use a variety of fonts, so we had to provide a way to load fonts from disk and cache them in memory. We also had to load hunks of code called "drivers" in a similar fashion, so Larry Kenyon and I collaborated on some code to load and cache objects from disk that we called the "object manager" that was used by the font manager.

Bill Atkinson had recently given Quickdraw the ability to scale bitmaps, including text, so we added support in the font manager for scaled fonts, which included heuristics to find the font that best satisfied a given request, if none were perfect. If we didn't have a font at the requested size, we could make one by scaling. But scaled text looked kind of lumpy, so we added a way for an application to choose if it wanted it or not.

Meanwhile, Bruce Horn was busy implementing the resource manager. I should have realized it sooner, but by the fall of 1982, it was becoming obvious that there would be lots of benefits if fonts and drivers became resources. We'd save space by discarding our "object manager" code, and simplify the system by eliminating lots of separate files; applications could even transparently contain their own fonts if they wanted to. I devised a simple scheme to encode the font family and size into a resource ID, and rewrote the font manager to be based on resources. Resource IDs were 16 bits long, with the high bit reserved, so the font manager ended up supporting a universe of 128 font families in 128 different sizes, which seemed like plenty at the time.

We added a number of refinements to the font manager in 1983, like a routine to make it easy to build the font menu, but the basic design didn't change very much, and that's what we ended up shipping with in January 1984. The software team was exhausted from the final push, and not all that much development happened during 1984, as some of the old-timers like myself left the company (I went on leave of absence in March 1984) and new hires like Burt Sloan and Ernie Beernick got up to speed. I certainly didn't think I'd be working on the font manager anymore.

In early 1985, Jerome Coonen was getting the software team revved up to do a 128K version of the ROM, that would eventually ship with the Mac Plus in January 1986. It incorporated lots of bug fixes, and some newer subsystems like Appletalk and a better, hierarchical file system. There was so much extra space available that it even incorporated a ROM-based resource file for system resources, to speed up booting and save memory. Jerome solicited my suggestions for improvements, and I responded with a pretty long list.

Among other things, I suggested a major overhaul of the font manager. The LaserWriter, with its superb Postscript imaging system, had made the font situation on the Macintosh pretty complicated, with different subsystems for the screen and printing, and the limitations of the font manager made it hard to get things right. We needed to support fractional pixel widths so the screen could better match the printer, and provide a better scheme for mapping fonts to resources that wasn't limited to 128 fonts. But I was disappointed that the team didn't decide to implement any of my font manager suggestions because development time was growing short and it seemed like too much work.

Even though I was no longer an Apple employee (see Things Are Better Than Ever) , I was still in close touch with the company, since Burrell Smith was my best friend and next door neighbor (we had bought two houses on the same lot in Palo Alto in April 1983). Bud Tribble had finished medical school and returned to Apple in his old role of software manager in August 1984, and was living in a spare room at Burrell's house, so I saw him frequently as well. And every once in a while, without advance warning, Steve Jobs would show up at my doorstep for an impromptu visit on a weekend afternoon.

On one such visit, in late February 1985, Steve asked me what I thought of the 128K ROM effort. I complained that it wasn't ambitious enough, and mentioned the font manager changes when Steve asked for an example. But I didn't expect his response.

"If you think that's so important, why don't you go ahead and do it yourself", Steve told me. "There would be time to get it done if you pitched in. Maybe we could give you a few Macs in exchange for doing it."

I told him I'd think about. Bud Tribble came by the next day to discuss it with me, and I agreed to develop a new font manager for the 128K ROM in exchange for three 512K Macintoshes, which I had Apple ship directly to my brothers and sister as gifts. All the code for the new ROM was supposed to be completed within a month, so I'd have to drop what I was doing to get the font manager done in time.

Pretty soon, I had the new font manager going, which used a new resource type called "FOND" to describe all of the properties of a font family, including the resource IDs of the font bitmaps. This allowed us to support tens of thousands of font families instead of a mere 128 like the old design. I also got the fractional width support going, but I needed some changes in QuickDraw to make it so applications could actually use it.

I talked with Bill about implementing the required changes and was surprised to find that he was reluctant to do it. In fact, even though we had identified a number of bugs, Bill didn't want to change QuickDraw at all, arguing that any sort of change at all would mess up existing applications.

Finally, while I was down at Apple to attend a coordination meeting, Bud, Steve Capps and I all ganged up on Bill, telling him that if he didn't want to maintain QuickDraw, he'd have to let one of us do it. Bill said he'd think about it overnight, and we planned to meet again the next afternoon to decide what to do.

We were all surprised when Bill showed up at Apple the next day, excited to give us a demo of ovals drawing almost twice as fast as they used to. Apparently, when he started looking at the code yesterday evening he got excited about working on QuickDraw again. He started to fix the bugs we identified, and one thing led to another, and he soon he saw a half dozen ways to improve things. I was happy that Bill was excited about QuickDraw again, but a little bit afraid that he had swung too far in the opposite direction.

At this point, there was only around a week left before code freeze, but it seemed like the font manager was just about finished. Bill implemented measuring with the fractional widths, and it seemed to work great. But then, the evening before we were going to freeze the code, I got a call from Bill at around 8pm.

"Hey, Andy, I've got some great news. I can speed up text drawing by more than 40%! But I need a little help from the font manager."

It turns out that Bill had realized that, on the average, since most text was lower case, characters only used around half of the total height of the strike bitmap - only a few characters had descenders, and most didn't go all the way to the top, either. QuickDraw usually could just skip the blank parts in the characters, saving lots of time.

QuickDraw didn't have time to measure the tops and bottoms of the characters on the fly, but it was a perfect job for the font manager, which could do it once, right after it loaded a new font into memory. For each character, it calculated an offset and a length, telling Bill where to start and stop the drawing. Bill told me the format of the table that he'd like the font manager to supply, and asked me if I thought I could do it.

I thought that it was crazy to attempt something like this with the freeze scheduled in less than 16 hours, but it was such a cool idea that I told Bill that I'd give it a try. I went out for dinner, bought two six packs of Diet Coke, and started implementing the font measuring code around 10pm. I came up with a nice, efficient way to do it, iterating through the strike bitmap a longword at a time, and, eight diet Cokes later, I thought I had it working just about the time the sun was rising; I even had enough time to sleep for a few hours before meeting Bill at Apple at 10am.

We crossed our fingers when we tried it out, but luckily both of our parts worked the first time, with Bill's tests showing that we achieved the projected 40% speed-up. Of course, the code freeze ended up slipping for other reasons, so we didn't really have to rush like we did, but the urgency made it a little bit more fun, I guess.

# The End Of An Era

**Author:** Andy Hertzfeld

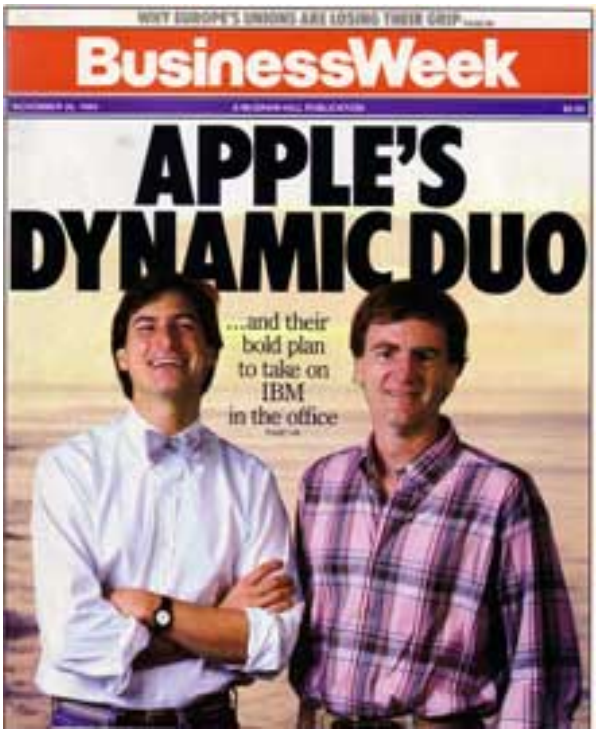
**Date:** May 1985

**Characters:** Bob Belleville, Joanna Hoffman, John Warnock, Mike Scott, Bud Tribble, Steve Jobs, John Sculley, Steve Capps, Bill Atkinson, Burrell Smith, Mike Murray, Jean-Louis Gassee, Andy Hertzfeld

**Topics:** Management, Personality, Personality Clashes, Quitting

**Summary:** The Mac division undergoes an inconceivable reorganization

**Revision:** most recent of 31




**Steve Jobs and John Sculley in November 1984**

The original Macintosh enjoyed robust sales following its spectacular launch in January 1984 (see The Times They Are A-Changin') . Steve Jobs defined success as selling 50,000 units in the first 100 days, which was a high hurdle for a brand new computer with only a handful of applications available. In fact, Apple was able to sell more than 72,000 Macintoshes by end of April, and continued to ramp up to sell over 60,000 units in June 1984 alone.

I travelled to the 1984 National Computer Conference show in June 1984 with the Mac team, sharing a hotel room with Burrell Smith, even though I was on leave of absence (see Leave Of Absence) . Apple had assembled over a dozen small software developers who had written cool applications for the Macintosh, to display them at the trade show. Steve Jobs was ebullient, and thought that the sprouting applications and blossoming sales meant that we had turned the corner. When I ran into him on the floor of the show, he put his arm around my shoulder and exclaimed, "Look at all these applications! We did it! The Macintosh has made it!"

High sales spurred even rosier predictions for the upcoming holiday season. But as summer turned into fall, Macintosh sales began to decline. For a couple of months, the University Consortium (see What's A Megaflop?) kept volumes high by selling tens of thousands of low cost Macs to college students, but by Thanksgiving 1984, sales had slowed significantly. The marketing team forecast selling over 75,000 Macs per month for the important holiday season, but actually they didn't even break 20,000 units per month. In December 1984, the Apple II still accounted for about 70% of Apple's revenues.

As the new year dawned, Steve Jobs seemed oblivious to the slowing sales, and continued to behave as if the Macintosh was a booming, unqualified success. His lieutenants in the Macintosh division, which had swelled to more than 700 employees, had to deal with a growing reality gap, reconciling the ever-changing audacious plans for world domination emanating from their leader with the persistent bad news from the sales channel.

Meanwhile, the Macintosh engineering team had not been very productive. The Mac was crying out for an internal hard drive, and some kind of high bandwidth port to attach it to, but there weren't any significant upgrades on the horizon, even though the basic hardware hadn't changed (except for additional RAM) for a year. In the fall of 1984, Steve Jobs tried to rally the remains of the original Mac team around the "Turbo Mac" project, featuring a new digital board with custom chips and fast I/O for an internal hard drive to be designed by Burrell Smith. But Burrell felt that engineering manager Bob Belleville was flinging lots of gratuitous obstacles in his path, and it eventually became so frustrating that he quit the company in February 1985 (see Are You Gonna Do It?) .

The only upcoming new product was the LaserWriter printer, based on Canon's 300 dots/inch laser printing engine, with a digital board designed by Burrell Smith and software written by Adobe, a new company founded by Xerox alumni John Warnock and Chuck Geschke. Like the Macintosh itself, Adobe's Postscript software at the heart of the LaserWriter was years ahead of its time, and was capable of producing exquisitely beautiful pages. Unfortunately, the LaserWriter had one major flaw: its retail price was over $7,000, almost triple the cost of a Macintosh.

Joanna Hoffman, the Mac team's original marketing person, transfered from international marketing back to the main product marketing group in early 1985, to help deal with the growing crisis. At the first sales meeting that she attended, she was surprised to see that the sales forecasts for the upcoming quarter were unchanged from six months ago, when things were still looking good, and were almost four times what they were currently selling. Everybody was informally assuming more realistic numbers, but no one had the heart to cut the official forecast, because they were afraid to tell Steve about it. Joanna immediately slashed the forecasts, to the relief of the sales and manufacturing team.

The weak sales were beginning to put pressure on the relationship between Steve Jobs and John Sculley for the very first time. They had gotten along fine when everything was going well, but hitherto they never had to deal with much adversity. Unfortunately, in early 1985 the personal computer market was descending into one of its periodic downturns, and even Apple II sales were starting to falter. Steve did not take criticism very well, and sometimes reacted to suggestions for improving Macintosh sales as if they were personal attacks. Their relationship began to sour as John put pressure on Steve to address the Macintosh's problems.

Steve Jobs had never suffered fools gladly, and as the pressure mounted, he became even more difficult to work with. Employees from every part of the company began to approach John with complaints about Steve's behavior, including some of Steve's direct reports in the Macintosh Division. John felt especially strongly about building more compatibility bridges with the IBM PC, an approach which Steve disdained. John began to view Steve as an impediment toward fixing Apple's problems, and the board of directors were urging him to do something about it.

Steve had often professed that he preferred working with small teams on new products, and that he didn't really want to run a large organization with hundreds of employees. Apple's board felt that he should hand the reins of the Macintosh division over to a professional management team, and return to his core strength as a new product visionary.

Steve had recently met an interesting character named Steve Kitchen, who was introduced to him by Steve Capps. Steve Kitchen was a fast talking, enthusiastic entrepreneur who had developed a couple of successful Atari video games. He claimed to have recently invented a revolutionary flat screen display technology that could facilitate portable computers. Steve Jobs was intrigued by the prospect of developing a lightweight portable computer, years ahead of its time, and he considered having Apple buy the technology and start a research organization called "Apple Labs" to develop it. But he seemed ambivalent, sometimes enthusiastic about starting Apple Labs, but other times he seemed determined to prove that he could manage the large division.

The conflict came to a head at the April 10th board meeting. The board thought they could convince Steve to transition back to a product visionary role, but instead he went on the attack and lobbied for Sculley's removal. After long wrenching discussions with both of them, and extending the meeting to the following day, the board decided in favor of John, instructing him to reorganize the Macintosh division, stripping Steve of all authority. Steve would remain the chairman of Apple, but for the time being, no operating role was defined for him.

John didn't want to implement the reorganization immediately, because he still thought that he could reconcile with Steve, and get him to buy into the changes, achieving a smooth transition with his blessing. But after a brief period of depressed cooperation, Steve started attacking John again, behind the scenes in a variety of ways. I won't go into the details here, but eventually John had to remove Steve from his management role in the Macintosh division involuntarily. Apple announced Steve's removal, along with the first quarterly loss in their history as well as significant layoffs, on Friday, May 31, 1985, Fridays being the traditional time for companies to announce bad news. It was surely one of the lowest points of Apple history.

I was shocked when I heard the news that morning from a friend at Apple, and immediately drove down to Cupertino to see what was going on, and commiserate with my friends. I was aware of the problem with Macintosh sales, but it was still inconceivable to me that the board could oust Steve Jobs, who was clearly the heart and soul of the company, difficult as he may sometimes be. It was almost impossible to imagine the Macintosh team without him at the helm. I thought that perhaps I wasn't hearing the whole story, and that something would emerge to help it make more sense.

I arrived at the Apple campus soon after Sculley's communication meeting finished, where he explained the nature of the reorganization and the accompanying layoffs. The way that people were milling around listlessly reminded me of Black Wednesday four years earlier (see Black Wednesday) , when Mike Scott unexpectedly purged the Apple II group. A few folks from the Apple II division who resented Steve's superior attitude seemed elated, and a few others saw the shake-up as an opportunity for personal advancement, but most of Apple's employees were sombre and depressed, feeling sad and uncertain about the future.

Lots of people had varying stories about what had actually happened. I thought that maybe it meant that Steve had decided to pursue AppleLabs, and that maybe I could come back to Apple to work on a small team again. I was anxious to talk to Steve himself, and find out his take on it, and I wasn't the only one. Bill Atkinson, Bud Tribble, Steve Capps and myself arranged to visit Steve at his house in Woodside for dinner on Sunday evening, two days after the reorganization was announced.

I had never been to Steve's house in Woodside before. It was a 14-bedroom, 17,250 square foot Spanish colonial style mansion built in 1926 that Steve had purchased around a year ago, in 1984. We knocked on the door and waited a few minutes before Steve appeared and led us inside. The massive house was almost completely unfurnished, and our footsteps echoed eerily as he led us to a large room near the kitchen, with a long table, one of the few rooms that had any furniture.

We stood around the kitchen chatting, as Steve prepared some food. His girlfriend Tina was there, who I had met a few times before; I was impressed by her mix of kindness and intelligence. Bill started chatting with Tina as I finally got a chance to ask Steve about the reorganization.

"So what really happened at Apple?", I asked him, even though I was scared to bring it up so directly. "Is it really as bad as it looks?"

"No, it's worse", Steve replied with a pained expression. "It's much worse than you can imagine."

Steve was adamant about blaming John Sculley for everything that had happened. He felt that John had betrayed him and he had little faith that Sculley or anyone else could manage Apple without him. He said that his role as chairman was completely ceremonial, and it left him with no actual responsibilities. In fact, Apple had already moved his office from Bandley 3 to Bandley 6, a small building across the street that was almost empty. The new office was so remote from day to day operations that it later was nicknamed "Siberia".

We had a pleasant dinner, huddled around one end of the long table, mainly reminiscing about the good old days developing the Mac but occasionally engaging in grim speculation about Apple's future. Steve had arranged for some gourmet vegetarian food to be delivered, and we drank some excellent wine. Dessert consisted of handfuls of locally grown Olson's cherries, grabbed from a large wooden crate that Steve kept in the kitchen.

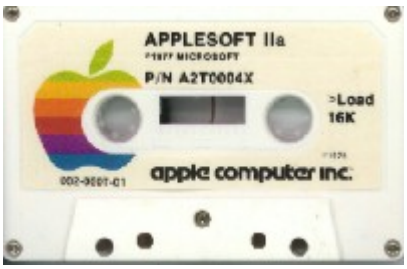
After dinner, we retired to another room that had an expensive stereo system and an elaborate model of the mostly underground house that Steve planned to build to replace the one we were standing in. I had brought along a copy of Bob Dylan's new album with me, "Empire Burlesque", which was just released earlier that week, because I knew that Steve, like myself, was a big Bob Dylan fan, although Steve thought that Dylan hadn't done anything worthwhile since "Blood on the Tracks" a decade ago. I placed the album on a hi-tech turntable that seemed to be mounted on aluminum cones and played the last song, "Dark Eyes", which was slow and mournful, with a sad, fragile melody and lyrics that seemed relevant to the situation at Apple. But Steve didn't like the song, and wasn't interested in hearing the rest of the album, reiterating his negative opinion of recent Dylan.

Later, when it was time to leave, we lingered outside under the beautiful summer night sky. We were all pretty emotional by then, especially Steve. I tried to convince him that the change wasn't necessarily so bad, and that I would be excited about returning to Apple to work with him on a small team again. But Steve was inconsolable, and more depressed than I had ever seen him before. As we left, I thought that it was lucky that he had Tina there to keep him company in the cavernous mansion.

It took a while for me to understand the consequences of the reorganization. The best news for me was that my nemesis Bob Belleville had resigned from Apple, because he had sided with Steve during the recent infighting and burned too many bridges to continue. Most of the rest of Steve's staff stayed on to work for Jean-Louis Gassee, who replaced Steve in the reorganized division, although Mike Murray resigned soon thereafter. Steve Jobs spent most of the summer traveling, trying to figure out what to do next. He was still the chairman of Apple Computer, but he was so at odds with the rest of its leadership that it was hard to see how he could remain there much longer.

# MacBasic

**Author:** Andy Hertzfeld
**Date:** June 1985
**Characters:** Donn Denman, Bryan Stearns, Jerome Coonen, Randy Wigginton, John Sculley, Bill Gates, Bill Atkinson, Steve Wozniak
**Topics:** Software Design, Management
**Summary:** The sad story of MacBasic
**Revision:** most recent of 4




**Applesoft Basic Cassette**

When the Apple II was first introduced in April 1977, it couldn't do very much because there were few applications written for it. It was important to include some kind of programming language, so users, who were mostly hobbyists, could write their own programs. "Basic", which was designed for teaching introductory programming by two Dartmouth professors in the 1960s, became the language of choice for early microcomputers because it was interactive, simple and easy to use. The Apple II included a Basic interpreter known as "Integer Basic", written from scratch by Steve Wozniak, which was almost as idiosyncratically brilliant as his hardware design, stored in 5K bytes of ROM on the motherboard. It also came with Microsoft's Basic interpreter, dubbed "Applesoft Basic", on cassette tape. Sadly, Applesoft eventually displaced Integer Basic in ROM in the Apple II Plus because it had the floating point math routines that Woz never got around to finishing.

Donn Denman started working at Apple around the same time that I did, in the summer of 1979. His job was to work with Randy Wiggington on porting Applesoft Basic to the Apple III. They needed to rewrite parts of it to deal with the Apple III's tricky segmented memory addressing, as well as porting it to SOS, the new operating system designed for the Apple III. It was easy for me to track Donn's steady progress because he sat in the cubicle across from mine when we moved into Bandley III in the spring of 1980.

By the summer of 1981, the Macintosh project was beginning to hit its stride, and we started thinking about the applications that we wanted to have at launch to show off the unique character of the Macintosh. Besides a word processor and a drawing program, we thought that a Basic interpreter would be important, to allow users to write their own programs. We decided we should write it ourselves, instead of relying on a third party, because it was important for the Basic programs to be able to take advantage of the Macintosh UI, and we didn't trust a third party to "get it" enough to do it right.

I still had lunch with some of my friends in the Apple II group a couple of times a week, and I started trying to convince Donn to join the Mac team to implement our Basic. He was reluctant at first, since the Mac project was still small and risky, but he was pretty much finished with Apple III Basic and was full of ideas about how to do it better. He eventually couldn't resist and joined the Mac team in September 1981.

A Basic interpreter consists of a text editor for inputing your program, a parser to translate it into a series of byte codes, and an interpreter to execute the byte-coded instructions. Donn wrote the interpreter first, and then hand-coded some byte codes to test it. He implemented some graphics primitives early on, since they were nice to demo. In a few months, he had a pretty impressive demonstration program going that drew elaborate graphical trees recursively, in multiple windows simultaneously, showing off the interpreter's threading capabilities.

By the spring of 1982, it was apparent that Donn needed some help if we wanted Basic ready for the introduction, which at the time was supposed to be in January 1983. We decided to hire Bryan Stearns to help him, who Donn knew from the Apple II team. Bryan was only 18 years old, but he was excited about the project and Donn thought they worked well together, so we gave him a chance.

But Basic still had a hard time getting traction, especially since the system was evolving rapidly beneath it. After six months or so, I was surprised to hear that Bryan was quitting the project to work at a tiny start-up founded by Chuck Mauro, who I had helped with his 80 column card for the Apple II. I tried to talk him out of it but he left anyway. By the spring of 1983, it was so obvious that Basic wouldn't be ready for the introduction that the software manager, Jerome Coonen, pulled Donn off of it to work on other parts of the ROM and the system. Donn worked on desk accessories and wrote the alarm clock and notepad, as well as the math guts of the calculator (see Desk Ornaments) .

After the Mac shipped in January 1984, Donn went back to work on Basic with renewed vigor, determined to get it finished. Apple brought in some free-lance writers to write books about it (including Scot Kamins, who was a co-founder of the first Apple users group in the Bay Area). But Microsoft surprised us, and released a Basic for the Macintosh that they didn't tell us they were developing. It was everything that we expected and feared, since it was essentially console-based - it didn't really use the Mac user interface. Donn was making good progress and looked to be on track to ship in early 1985; we were excited to show the world what Basic should really look like on the Macintosh.

Unfortunately, there was another problem on the horizon. Apple's original deal with Microsoft for licensing Applesoft Basic had a term of eight years, and it was due to expire in September 1985. Apple still depended on the Apple II for the lion's share of its revenues, and it would be difficult to replace Microsoft Basic without fragmenting the software base. Bill Gates had Apple in a tight squeeze, and, in an early display of his ruthless business acumen, he exploited it to the hilt. He knew that Donn's Basic was way ahead of Microsoft's, so, as a condition for agreeing to renew Applesoft, he demanded that Apple abandon MacBasic, buying it from Apple for the price of $1, and then burying it. He also used the renewal of Applesoft, which would be obsolete in just a year or two as the Mac displaced the Apple II, to get a perpetual license to the Macintosh user interface, in what probably was the single worst deal in Apple's history, executed by John Sculley in November 1985.

When Donn found out that MacBasic had been cancelled, he was heart-broken. His manager told him "it's been put on hold indefinitely" and instructed him to destroy the source code and all copies, but refused to answer Donn's questions about what was going on. Later that day Donn went for a wild ride on his motor cycle and crashed it, returning home scraped up but with no real damage, except to his already battered ego.

Bill Atkinson was outraged that Apple could treat Donn and his users so callously, and let John Sculley know how he felt, but the deal was done and couldn't be reversed.

Donn quickly filed for a leave of absence, but eventually returned to Apple to work on various projects, including AppleScript.

The Beta version of MacBASIC had been released to interested parties, including Dartmouth University which used it in an introductory programming class. Apple tried to get back all the copies, but the Beta version was widely pirated, and two books on MacBASIC were published, and sold quite well for several years.

# An Early 'Switch' Campaign

**Author:** Tom Zito
**Date:** September 1985
**Characters:** Bo Derek, Steve Jobs
**Topics:** Personality, Celebrities
**Summary:** Steve Jobs tries to get Bo Derek to abandon her PC in favor of a Mac
**Revision:** most recent of 9

**Bo Derek, PC User**

My former wife, Laura Bachko, had been a book editor in New York and had worked on a project with Bo Derek before we moved to California. One night when Laura and I were having dinner with Steve, Bo's name came up -- as did the factoid that Bo was a heavy duty computer jockey, albeit of the IBM persuasion. Steve took this as a personal challenge; he was going convert Bo to a Mac user -- and who knew what else. Clearly the computer could be a foot in the door. He persuaded Laura to make him an appointment with Bo.

And so one day shortly thereafter, Steve piled into his Mercedes, along with a Mac, and drove down to Bo's Santa Barbara ranch, which she shared with her husband John Derek. Bo was cordial but unimpressed; she accepted the computer but remained a PC user. And apparently she did not find Steve as dashing as Steve expected she would.

Several weeks later, Steve was complaining to Laura about the lackluster impression he had made. "Look," she told him. "She's married. And besides, I don't know any woman who would want her name to be Bo Jobs."

# Vault of Horror

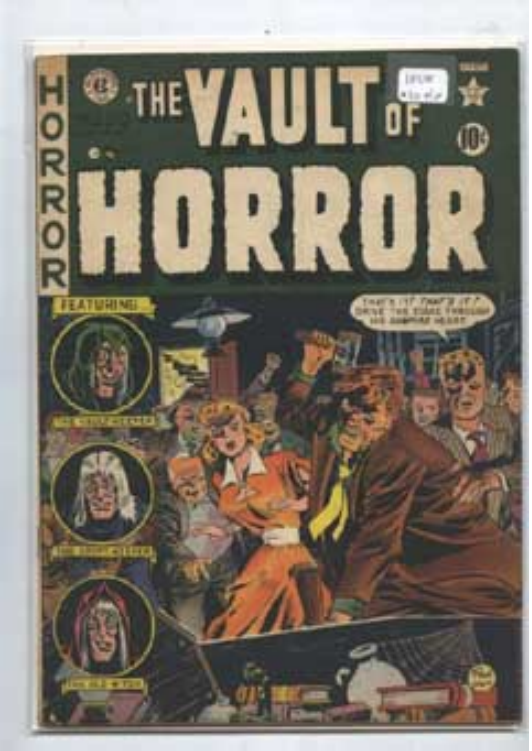
**Author:** Caroline Rose
**Date:** October 1985
**Characters:** Caroline Rose, Bill Dawson, Scott Knaster, Chris Espinosa
**Topics:** Documentation
**Summary:** The Inside Mac cover you never saw
**Revision:** most recent of 15

**Proposed Inside Mac Cover**

Scott Knaster, who was in Apple's developer tech support group when I was working on Inside Macintosh (the Mac developer documentation), has a great sense of humor and is a comic book fan. So it was no big surprise when, shortly before the publication of the combined hardcover edition of Volumes I through III in November 1985, he gave the Inside Mac team a "proposed cover" for that edition that was based on the cover of an old comic book called the Vault of Horror. Scott had given a reprint of issue 20 of this comic (originally published in 1951) to Bill Dawson in Apple's production group and asked him to alter it in certain ways to make it apply to Inside Mac. Bill did this by cutting and pasting with scissors and glue, because in those days there were no other options, like scanning it in and doctoring it in Photoshop. I've included the original along with the doctored version in this story, so you can compare.

We cracked up at the parts of the Macintosh Toolbox Scott chose as the names of the three evil-looking characters stacked up on the left: the Control Manager, the Segment Loader, and the Trap Dispatcher. The coffin containing the vampire whose heart is having a stake driven through it was labeled "BIG BLUE" to show how Apple was going to wipe out IBM with the success of the Mac. More subtle was the change made to what's being said to the man driving the stake: he's addressed as "Ed" because the group we were in was called User Ed (short for User Education). And in case the $100 price Bill added looks like a joke: it wasn't; it was an educated guess at what the humongous hardcover edition would cost (which ended up at $79.95).

An easily overlooked change appears in the upper left, in the "EC" logo: where it said "Entertaining Comics," Bill changed it to "Espinosa Comics," reflecting Chris Espinosa's role as the manager of User Ed.




**Original Comic Book Cover**

Apple of course didn't use this cover, but six years later, during my stint as editor in chief of *develop,* Apple's technical journal for Mac developers, I thought it would be fun to reprint the cover in an issue of *develop.* From the reverse of the cover I saw that the original comic was published by William M. Gaines, whose name I was familiar with as the eccentric publisher of MAD magazine. Gaines had been a hero of mine since childhood; in fact, I billed myself as the "editor in cheek" of *develop* and gave the journal as playful and irreverent a tone as I could get away with. So I contacted his office in New York and asked for permission to publish the altered cover. A few weeks later I received a call from someone who, in a tone that was more morose than professional, said he was William Gaines and was granting me the permission I requested. That the man himself should call me was quite a surprise and a thrill—and I'm glad he didn't procrastinate, because he died not much later (in June 1992).

It turns out that there were people at Apple who objected to publishing the cover in *develop.* I think that by then they thought driving a stake (however metaphorical) through IBM's heart would be impolitic. I'm happy that now I finally have a chance to show it to a wider audience.

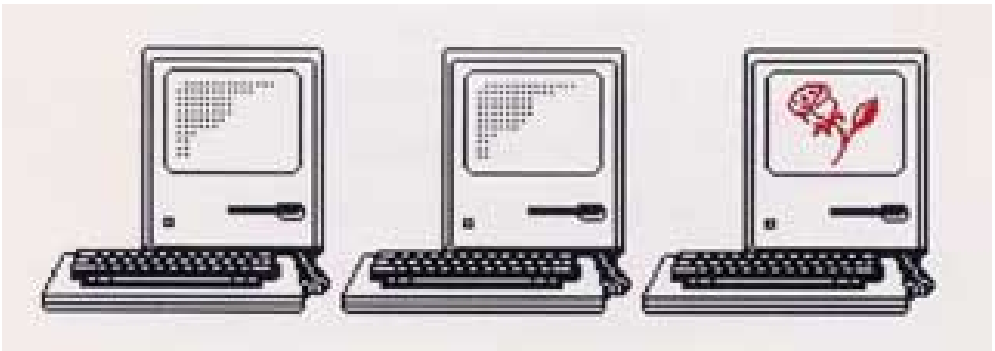
# A Rose by Any Other Name

**Author:** Caroline Rose
**Date:** December 1985
**Characters:** Caroline Rose, Bob Anders, Brad Hacker, Dan Cochran, Louella Pizzuti
**Topics:** Documentation, Credit
**Summary:** The Easter egg in Inside Macintosh
**Revision:** most recent of 14

When the Inside Macintosh developer documentation was finally published, it reflected years of work I'd done along with Bob Anders, Brad Hacker, and a few others. I'd guided the style of it, written most of it, and edited all of it. More recently, I'd supervised a nutty, charming young woman named Louella Pizzuti who was hired to convert the files from the Apple III over to the Macintosh. (Nearly six years later, she hired me to take over *develop,* Apple's technical journal, but that's another story.) Suffice it to say my life had been consumed by Inside Mac for three and a half years.

The product manager, Dan Cochran, oversaw Inside Mac through its production into print. He faced some difficult decisions regarding how to publish the first edition, and eventually it was decided that, in addition to separate softcover editions of Volumes I through III, the three volumes would be combined into one huge hardcover edition. (We jokingly referred to it as a coffee table book: big enough to use as a coffee table.) A combined preliminary "phone book" edition on flimsy paper had already been distributed, but I was of course eager to see the final printed version.

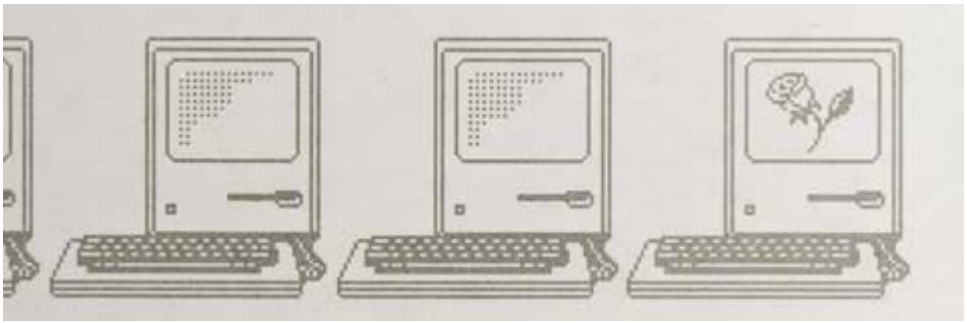
The hardcover edition was actually in hand by the end of November 1985—but not in my hands: I was told I couldn't see it until a celebratory event that was scheduled for December 4. People at Apple aren't usually big on formalities, but I could understand wanting to make a ceremony of this occasion. I was delighted to see that one of the three pixelated Mac images on the printed invitation to the event showed a red rose on its screen. I was often called "C. Rose" by others on the team, and I made no secret of how much I liked (and identified with) that flower.
Seeing the rose on the invitation brought a big smile to my face.



**From the invitation to the Inside Mac event**

The event started with a few words about the Inside Mac effort, followed by a presentation to team members of crystal bookend-like plaques etched with an image of the Mac (the same as on the invitation, sans the rose) and a personal inscription. But the highlight of the ceremony for me was being handed the hardcover edition itself. Everyone else seemed all atwitter over this, I figured because they knew what a highly anticipated moment it was for me.

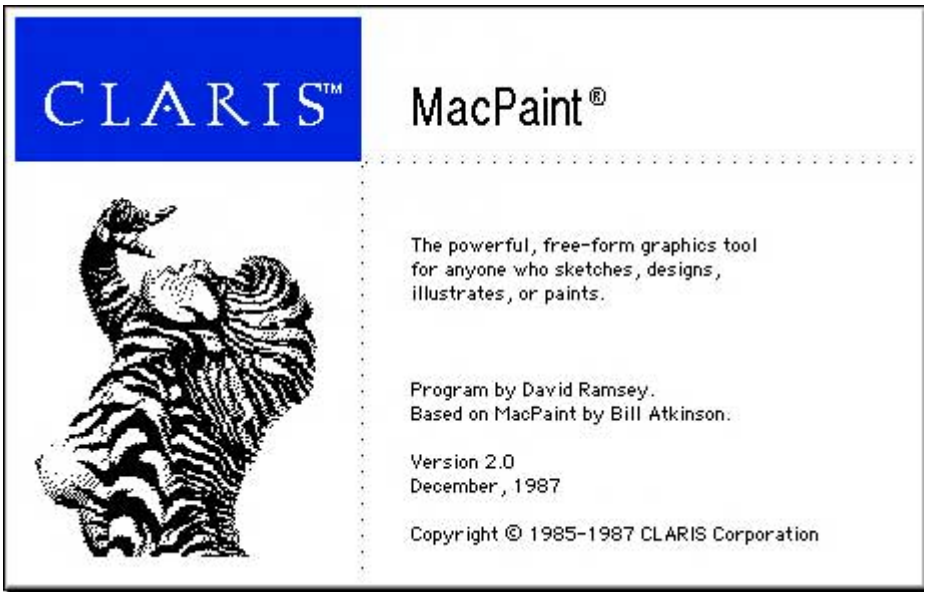
Upon opening the book, I saw that a row of Macs like the one on the plaque had been printed on the reverse of the front cover, all along the bottom, as well as on both sides of a stiff sheet inserted before the title page. As I began to thumb through the book, the others around me seemed impatient, and finally asked me to turn to the inside of the back cover; not surprisingly, the now-familiar row of Macs appeared there and on the sheet inserted before it. Then they asked me to lift up the dust jacket so I could see the last couple of Macs in the row, and that was when I saw it: a rose in the last Mac. Seeing *that* rose brought tears to my eyes, and made it a day—and a tribute—I would always remember. (I also received a small rhododendron bush from Dan, which is now flourishing in front of my house, so every Spring I have real flowers as an additional reminder.)



**From the inside back cover of the hardcover edition**

# Evolution Of A Classic

**Author:** David Ramsey
**Date:** March 1986
**Characters:** Bill Atkinson, Gene Pope, David Ramsey, Rich Page
**Topics:** Maintenance, MacPaint
**Summary:** MacPaint 2.0
**Revision:** most recent of 17




**MacPaint 2.0's**

I was hired at Apple in early 1986. Oddly, I was given nothing to do when I started, so after setting up my cube and getting my Lisa (all Mac programming was done on the Lisa back then) running, I wandered over to my manager's cube and asked if he had the source to MacPaint. He had-- he handed me a single Lisa diskette with all of MacPaint on it.

MacPaint was the first Mac program I'd ever seen; Bill Atkinson was demonstrating it at Computers Plus, a local Apple store run by Mark Wozniak and Dick Applebaum. I was completely awestruck at how fast the computer could "sling bits" on the screen. But there were still things I wanted to change: I wanted to be able to move and resize the image window, for example.

After a week or so of playing around I showed my manager (Gene Pope) what I'd done. I had just been killing time, since nobody had told me to do anything, but Gene asked me to add more features with an eye towards a "MacPaint 2.0".

**And so it begins**

At first I was nervous: Bill Atkinson is a hard act to follow, and I was also worried what he might think about some newbie hire taking over his baby. But he was tremendously supportive (and it probably also helped that he was deeply involved with his new project, Hypercard) and even took hours of time to go over his hand-optimized 68000 assembly language routines to explain how they worked. I still remember him peering intently at me, making sure I understood the importance of what he was saying: "Unroll the loops", "keep the registers full"...

The day I finally figured out how "regions" in QuickDraw worked was an epiphany, really.

**Specs? We don't need no steenking specs!**

It seems strange by modern standards, but MacPaint 2.0 never had a product spec. There was no marketing requirements document. There were no design meetings. There wasn't even a product manager until the last few months of work. Many people had opinions and suggestions (and the

then-new tear-off menus came directly from HyperCard, courtesy of Bill) but ultimately the product was what *I* wanted it to be.

Of course, I had more leeway than Bill did with the original MacPaint since I didn't have to worry about running on a 128K Mac-- a 512K computer was the minimum. Thus MacPaint gained a sizeable, movable painting window; multiple document support; large clipboards, the "snapshot" capability with the "magic eraser", and so forth.

Some features were pretty subtle: for example, in response to a request from Radius, MacPaint 2.0 would automatically move its window down if the user took advantage of a Radius monitor's capability to enlarge the standard menu bar!

**Fun hardware note**

Most of MacPaint was developed on a "Big Mac" prototype-- a computer that was a design study for the next Macintosh. Basically it was a 16mHz 68020 version of a Mac Plus. Cases were never made, so it was simply a 1 foot square circuit board mounted on a piece of wood, connected to a 10 megabyte SCSI hard drive.

I used the Big Mac prototype since it was faster and more reliable than the Macintosh II prototypes available. It was never produced, and designer Rich Page left Apple to work at NeXT shortly after his design "lost" to the slotted Mac II.

I still have the Big Mac I used at Apple. I wonder if it still works.

**The Zebra Lady**

Test versions of MacPaint had different images, generally from comic books, in the "About" box. The last beta version showed the nude upper torso of a zebra-striped woman taken from the Olivia de Berardinis painting "Zebra Lady". With the covert approval of all involved, this version made it into the release: you could show the Zebra Lady image by holding down the tab and space keys while selecting "About MacPaint".

When the artist found out, she took it well. I have a signed copy of the original print framed to this day.

**Shelf Life**

MacPaint 2.0 was introduced in late 1987 (did it really take me that long to write?) and remained on the market until fall 1998. I think this is a record of some sort for personal computer software.

# A Lesson In Diplomacy

**Author:** David Ramsey

**Date:** 1988

**Characters:** David Ramsey, Jean-Louis Gassee

**Topics:** Marketing, Hardware, Mac II

**Summary:** What's worse than a dealer screwing up your new computer? Telling the customer the dealer screwed up their new computer.

**Revision:** most recent of 5

While at Apple, I was active online in the Compuserve Macintosh Forums. I was known as an Apple employee there and provided what support I could to people with problems or questions.

One person had a particularly vexing problem. He'd purchased a fully outfitted Macintosh II computer, with two megabytes of dealer-installed RAM and a 40 megabyte hard disk, video card, and Apple color monitor. At the time this configuration cost over $5,000.00.

He claimed the computer had never worked properly-- that it was flakey and crashed frequently. He had returned it to his dealer multiple times and had finally been told that the dealer couldn't do anything else for him and not to come back.

I suggested several things for him to try, none of which worked. I was puzzled since by that time I was using a Mac II as my work machine and knew it to be solid and reliable.

Finally I suggested that he ship the computer to me at Apple, where I promised I would get to the bottom of his problems. When I received his computer, it was indeed flakey and crashed frequently. When I looked inside his computer the reason for the problems became obvious...

Macintosh II computers had 8 slots for SIMM memory modules. To install memory, the SIMM had to be inserted into the socket at an angle and then pivoted into place until a pair of plastic locking pins snapped around it. To remove the SIMM, the plastic locking pins had to be carefully pried apart so the SIMM could be tilted out of the socket.

It was very easy to break the locking pins when removing SIMMs. This resulted in a SIMM that wasn't held securely in its socket.

When I opened the customer's computer, I saw several of the SIMM sockets' locking pins had been broken. Whoever had caused the damage had attempted to hold the SIMMs in place with rubber bands. Since the computer had been purchased new and the memory installed by the dealer, it seemed obvious to me that the dealer had damaged the computer and was trying to duck his responsibility to repair or replace it.

Fortunately I had a friend who worked at the Apple service center in Campbell (the building, at the intersection of Highway 17 and Hamilton, is now a Fry's Electronics). I removed the Mac's logic board and took it over to the service center, where my friend used industrial soldering equipment to remove the damaged sockets and replace them with new ones. I reassembled the computer, reinstalled the memory, and set it running Apple diagnostic software. After a successful 24 hour run I returned the computer to the customer...with a letter noting what the problem was, how it was repaired, and oh, by the way, it was the dealer's fault.

It's been a long time but I think I even went so far as to comment that even if the dealer didn't do it, it was pretty hard to explain how he'd never noticed the broken SIMM sockets.

The customer showed the dealer my letter. This landed me in some hot water, as the enraged dealer called Apple demanding my head. He didn't get it (then). Jean-Louis Gassee took me aside to explain that it was great that I'd taken an unhappy Apple customer and made him a happy Apple customer. If all I'd done, he said, was to repair the computer and enclose a letter explaining the problem, nobody could have criticised me. But fingering the dealer (which he personally thought was correct) was the wrong thing to do: he noted that I had only the customer's unsupported word on the events leading to the problem, and that if the customer's account of the matter were accurate, he'd easily figure out it was the dealer himself.

As far as I know, no action was ever taken against the dealer. But Gassee's explanation was of how to handle situations like this was a valuable lesson.

# Origins of Spline-Based and Anti-Aliased Fonts

**Author:** William Donelson
**Date:** January 1989
**Characters:** William Donelson, John Warnock, Bob Sproull
**Topics:** Origins, Software Design
**Summary:** Creation of the first spline-based font system
**Revision:** most recent of 5

While working on a spring sabatical from high school in April and May of 1971, I worked in the computer department of the National Institutes of Health in Maryland.

I was assigned to work on the PDP-6 and PDP-10 with several people, including Dr Bob Sproull.

They had a Calcomp Film Plotter which used an electron beam to draw on film, thus allowing high resolution output, e.g. for publishing.

Bob asked me to investigate ways to make zoom-able fonts, which could also be distorted into a sort of italic, etc.

I studied the problem for a week or so, and concluded the best way to do this was a combination of straight lines and splines. I developed a list structure to store fonts. As examples, I did the letters "I" (of course) and "P" (just to prove the point).

This idea worked very well, and I gave a demo on the PDP-6's vector display system, set into raster mode. The characters could be scaled and slanted to "italic" quite easily, and could be shown as outlines or as filled characters.

(I also did a fsimple ighting video game on the display, "Germs" of course)

I then went off to MIT for ten years.

Fast forward to MacWorld, January 1989. I had taken a job at a Silicon Valley company doing Mac storage and video products, and was invited to a breakfast with John Warnock.

During breakfast, I mentioned to him that I was the originator of the spline-based font system. He looked shocked, and distinctly nervous! Apparently, he and Bob Sproull were associates, and he was afraid I was going to make claims or something!

I laughed, and put his mind to rest: "John, don't worry. I had an idea 18 years ago, and you and Adobe did all the real work". He laughed and relaxed a bit, but I'll never forget the expression on his face!

# Mea Culpa

**Author:** Andy Hertzfeld
**Date:** undated
**Characters:** Bruce Horn, Bud Tribble, Jerome Coonen, Bill Atkinson, Andy Hertzfeld, Bill Gates, Burrell Smith
**Topics:** Hardware Design, Software Design, Technical
**Summary:** Here are some of our worst mistakes
**Revision:** most recent of 12

Almost everyone involved with the design of the original Macintosh is proud of the work that we did on the project, both individually and collectively, but that doesn't mean that we aren't also embarrassed about some of the mistakes that we made. It's worthwhile to consider, if not apologize for, the worst decisions that I was personally responsible for, as well as other major faults in the system software and product as a whole.

The worst blunder that I perpetrated had to do with the memory manager. Bud Tribble adapted the Lisa intra-segment memory manager for the Macintosh (see Hungarian) , but we needed to add a few features. One was a "locked" attribute associated with a relocatable memory block, that temporarily prevented the block from being moved. Another enhancement was a "purgable" attribute, that told the memory manager that it could release a block if it needed to, as memory was getting full. The big mistake was where I chose to locate the bits that controlled the attributes.

I decided to put the bits controlling the "locked" and "purgable" attributes in the high-order bits of the master pointer (a pointer to the current address of a memory block), because they weren't being used for anything else. The 68000 had a 24 bit address bus, allowing 16 megabytes of addressable memory, so the high-order 8 bits of an address were not used by the processor. The high bit of a word is the most efficient one to test, which was another reason that I thought it was efficient to locate the flags there.

Of course, it was foolish to count on unused address bits to stay that way for very long, and it became a problem when the Macintosh transitioned to the 68020 processor in 1987, with the introduction of the Macintosh II. The 68020 had a full thirty-two bit address bus, so the memory manager could no longer get away with using the high-order master pointer bits for flags. It wasn't that hard for Jerome Coonen to convert the memory manager to keep the flags in the block header instead of the master pointer (which was where they should have been in the first place), but the practice of manipulating them directly had crept into third party applications, even though it wasn't supposed to, and it took another year or so to identify and eradicate all the transgressions to upgrade the Macintosh software base to be "32 bit clean", so the full address space could be used.

I paid a more direct price for my second worst mistake, which was to use fixed low memory addresses for toolbox globals. The Apple II kept important system globals in low memory, and the 68000 included a special 'short' addressing mode that made accessing addresses in the first 32K of memory more efficient, which motivated us to use low memory for various globals. While that may have been acceptable for system globals, it was a clearly a mistake for the toolbox, since that precluded us from running more than one application at a time, since each application required its own copy of the toolbox globals.

That didn't matter much at first, because with 128K of RAM, we barely had enough memory to run a single application at a time. But when the 512K Macintosh was released in September 1984, it started to become an issue. In October 1984, after I left Apple to work on my own, I realized that you could solve the problem by swapping all of the application-dependent low memory locations when you performed a context switch. In a few days, I wrote the core of the Mac's first multi-tasking environment called Switcher, using the low memory swapping technique, that kept multiple programs resident in memory at once, and switched between them with a nifty scrolling effect. Using low memory like we did ended up making context switching a few milliseconds slower than it should have been, and made it harder to eventually use a memory management unit, but it didn't turn out to be as devastating as I once thought.

We wanted the Macintosh to have a relatively simple system architecture, so it could perform well with limited hardware resources, but perhaps we went a little too far. We decided that we could live without a memory management unit, which was the right decision because of the expense of the associated hardware. But we also decided to eliminate the distinction between user and system code, by running everything in supervisor mode. This empowered applications and simplified the system, but it was a poor choice in the long run, because it made it harder to control the software base as the system evolved.

Even Bill Atkinson made an occasional error. His worst mistake was using signed 16-bit integers as sizes in various QuickDraw data structures like regions and pictures. This limited the maximum size of a region or picture to 32 kilobytes, which became a significant limitation a few years later as memory sizes grew. Bruce Horn's resource manager suffered a similar problem, using 16 bit offsets, limiting the size of resource files unnecessarily.

The biggest problem with the Macintosh hardware was pretty obvious, which was its limited expandability. But the problem wasn't really technical as much as philosophical, which was that we wanted to eliminate the inevitable complexity that was a consequence of hardware expandability, both for the user and the developer, by having every Macintosh be identical. It was a valid point of view, even somewhat courageous, but not very practical, because things were still changing too fast in the computer industry for it to work, driven by the relentless tides of Moore's Law. Burrell did try to sneak some expandability into the design (see Diagnostic Port) , but was only partially successful.

Limited hardware expandability exacerbates other flaws in the design, since you don't have the flexibility for yourself or third-parties to easily correct them. One of the biggest mistakes that we made in the first Mac was not enough support for a hard drive. Our first file system used a simple data structure that didn't scale well to large drives (in fact, it was suggested to us by Bill Gates in July 1981), and we didn't have a way to get bits in and out of the box at the rates that a hard disk required. In our defense, it was hard to for us to consider adding a hard disk to the Macintosh because it was one of the last differentiators from the Lisa, which was more than three times as expensive. But the lack of hardware flexibility made it more difficult for third parties to jump into the breach, although some did anyway.

From a broader perspective, I think that many of our mistakes came from a lack of understanding about exactly what we were doing. We thought that we were making a great product, reincarnating the Apple II for the 1980's, but we were actually creating the first in a long line of compatible computers that would persist for decades, although the latter wouldn't have happened if we didn't succeed at the former. Perhaps our design would have given the future more priority over the present if we had understood how long it would last.

# Busy Being Born

**Author:** Andy Hertzfeld
**Date:** undated
**Characters:** Bill Atkinson, Bud Tribble, Steve Jobs
**Topics:** Origins, User Interface, Early Programs, Lisa, QuickDraw, Software Design
**Summary:** A visual history of the development of the Lisa/Macintosh user interface
**Revision:** most recent of 30

The Macintosh User Interface wasn't designed all at once; it was actually the result of almost five years of experimentation and development at Apple, starting with graphics routines that Bill Atkinson began writing for Lisa in late 1978. Like any evolutionary process, there were lots of false starts and blind alleys along the way. It's a shame that these tend to be lost to history, since there is a lot that we can learn from them.

Fortunately, the main developer of the user interface, Bill Atkinson, was an avid, lifelong photographer, and he had the foresight to document the incremental development of the Lisa User Interface (which more or less became the Mac UI after a few tweaks) with a series of photographs. He kept a Polaroid camera by his computer, and took a snapshot each time the user interface reached a new milestone, which he collected in a loose-leaf notebook. I'm excited to be able to reproduce and annotate them here, since they offer a fascinating, behind the scenes glimpse of how the Mac's breakthrough user interface was crafted.

The images are scaled so they easily fit onto a typical screen, but you can click on them for larger versions that show more detail.

The first picture in Bill's notebook is from Bill's previous project, just before starting work on the Lisa: Apple II Pascal. The high performance graphics routines that Bill wrote for Apple II Pascal in the fall of 1978 led right into his initial work on the Lisa.

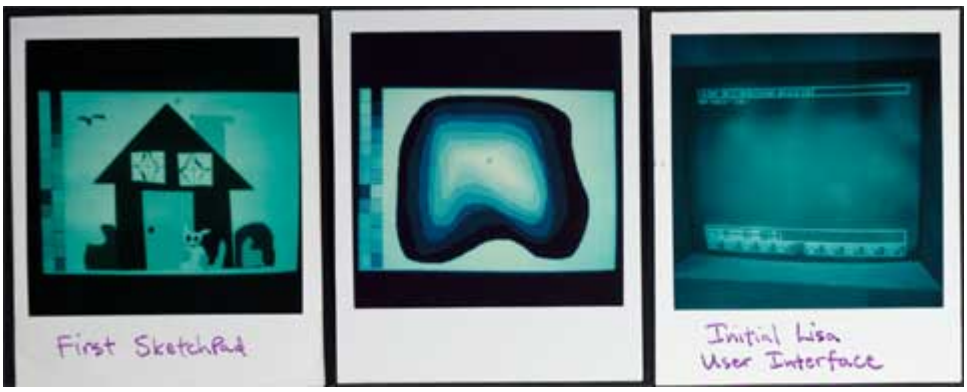
The center and right photos, from the spring of 1979, were rendered on the actual Lisa Display system, featuring the 720 by 360 resolution that remained constant all the way through to the shipping product. No Lisa existed yet; these were done on a wired wrapped prototype card for the Apple II. The middle picture shows the very first characters ever displayed on a Lisa screen; note the variable-width characters. The rightmost picture has more proportional text, about the Lisa display system, rendered in a font that Bill designed by hand.

The leftmost picture illustrates the first graphics primitives that Bill wrote for LisaGraf (which was eventually renamed to QuickDraw in 1982) in the spring of 1979, rendering lines and rectangles filled with 8x8 one-bit patterns. The power and flexibility of the patterns are illustrated in the rightmost shot, which were our poor man's substitute for color, which was too expensive (at the required resolution) in the early eighties.

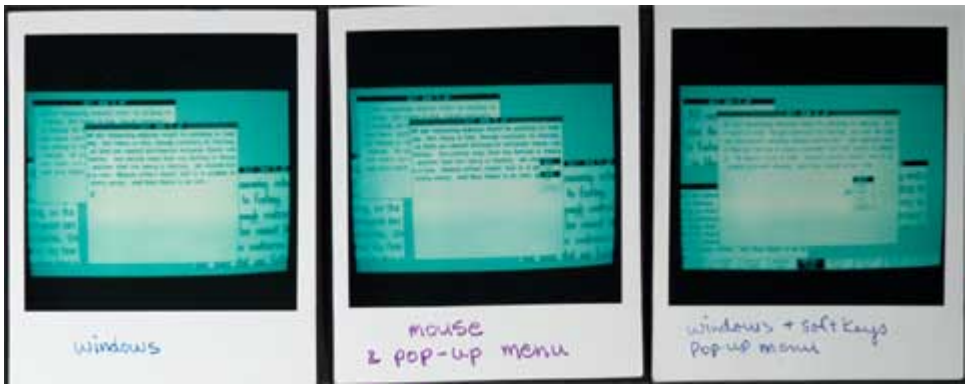
The middle picture depicts the initial user interface of the Lisa, based on a row of "soft-keys", drawn at the bottom of the screen, that would change as a user performed a task. These were inspired from work done at HP, where some of the early Lisa designers hailed from.

Here are some more demos of the initial graphics routines. Bill made line-drawing blindingly fast with an algorithm that plotted "slabs" of multiple pixels in a single memory access. The rightmost picture shows how non-rectangular areas could be filled with patterns, too.

Here are some scanned images, showing off Lisa's impressive resolution for the time, which Bill scanned using a modified fax machine. He was always tweaking the half-toning algorithm, which mapped gray scales into patterns of monochrome dots. Bill had made versions of these for the Apple II that Apple distributed on demo disks, but these higher resolution Lisa versions were much more impressive.




The left and middle pictures show off the first sketch program, an early ancestor of MacPaint, that allowed mouse-based drawing with patterns and a variety of brush shapes. I think these are perhaps a bit out of sequence, done in early 1980. The rightmost picture shows the final soft-key based UI, which is about to change radically...

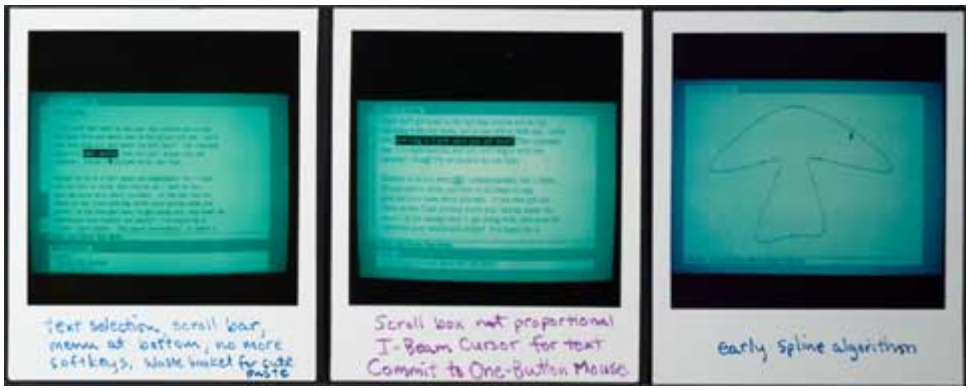



...into a mouse/windows based user interface. This is obviously the biggest single jump in the entire set of photographs, and the place where I most wish that Bill had dated them. It's tempting to say that the change was caused by the famous Xerox PARC visit, which took place in mid-December 1979, but Bill thinks that the windows predated that, although he can't say for sure.

The leftmost picture shows different fonts in overlapping windows, but we didn't have a window manager yet, so they couldn't be dragged around. The middle window shows the first pop-up menu, which looks just like SmallTalk, as does the simple, black title bar. The rightmost picture shows that we hadn't given up on the soft-keys yet.

By now, it's the spring of 1980 and things are starting to happen fast. The leftmost picture shows the earliest text selection, using a different highlighting technique than we ended up with. It also shows a "command bar" at the bottom of the screen, and that we had started to use non-modal commands (make a selection, then perform an action, instead of the other way around).

The middle picture shows the very first scroll bar, on the left instead of the right, before the arrow scroll buttons were added. It also has a folder-tab style title bar, which would persist for a while before being dropped (Bill says that at that point, he was confusing documents and folders). The right most photo shows that we adopted the inverse selection method of text highlighting.




By the summer of 1980, we had dropped the soft-keys. The leftmost photo shows that we had mouse-based text editing going, complete with the first appearance of the clipboard, which at that point was called "the wastebasket". Later, it was called the "scrap" before we finally settled on "clipboard." There was also a Smalltalk style scrollbar, with the scroll box proportional to the size of the document. Note there are also two set of arrows, since a single scrollbar weirdly controlled both horizontal and vertical scrolling.

The next picture shows that we dropped the proportional scroll box for a simpler, fixed-size one, since we were afraid users wouldn't understand the proportionality. It also shows the I-Beam text cursor for the first time. At this point, we were finally committed to the one-button mouse, after a long, protracted internal debate.

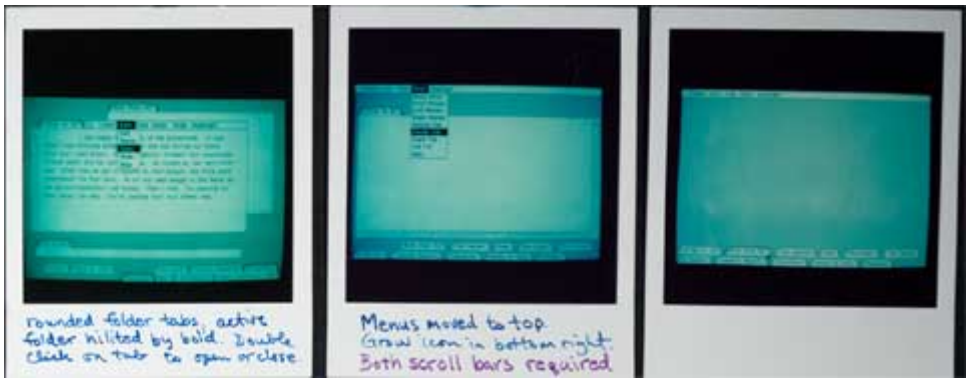
The right most picture shows Bill playing around with splines, which are curves defined by a few draggable control points. QuickDraw didn't end up using splines, but the picture is still notable for the first appearance of the "knobbie" (a small, draggable, rectangular affordance for a point).

By now, it's the fall of 1980. The middle picture shows us experimenting with opened and closed windows, which was eventually dropped (but it made a comeback in the 1990s and is in most systems today one way or another). The right most picture shows the first window resizing, by dragging a gray outline, although it's not clear how resizing was initiated.

The middle picture shows that windows can be repositioned by dragging a gray outline. We wanted to drag the whole window, like modern user interfaces do today, but the processors weren't fast enough in those days. As far as I know, the NeXTStep was the first system to do it the modern way.

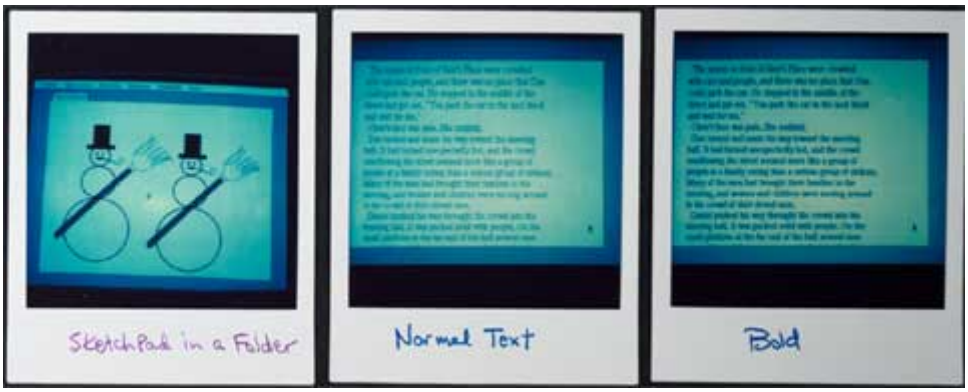
The right most picture shows the first appearance of pull-down menus, with a menu bar at the top of the window instead of the top of the screen, which is the way Windows still does things. By this point, we also gave up on using a single scroll bar for both horizontal and vertical scrolling; it's looking very much like what the Mac shipped with in 1984 now.

This set of pictures illustrates the Lisa desktop, circa the end of 1980, with a tab-shaped title, followed by a menu bar attached to the window. Windows could be reduced to tabs on the desktop. We've also changed the name of the clipboard to "the scrap", an old typesetting term.

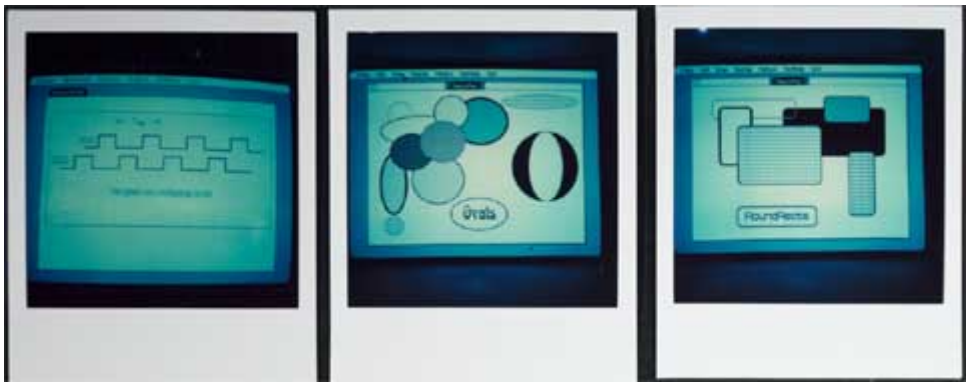



The leftmost picture mentions the first use of double-click, to open and close windows. The middle picture represents a real breakthrough, by putting the menu bar at the top of the screen instead of the top of each window. The menu bar contains the menus of the "active folder", which is the topmost window. By this point, the grow icon found its way to the bottom right, at the intersection of the horizontal and vertical scrollbars, which stuck. This is the first picture which is really recognizable as the shipping Macintosh.

By now, it's early 1981, and things are beginning to shape up. The leftmost picture shows a window with scrollbars that look a lot like the ones that shipped. The middle folder illustrates split views, which were used by Lisa's spreadsheet application. The rightmost picture contains the first appearance of a dialog box, which at the time ran the entire length of the screen, just below the menu bar.




Now that the basic window structure was stabilizing, Bill turned his attention back to the graphics routines. He worked more on the Sketch program (the forerunner of MacPaint); the snowman drawing on the left is a clue that it's now Winter 1981. He added algorithmic text styles to the graphics, adding styles of bold (pictured on the right), as well as italic, outline and shadow (Bill took pictures of the other styles which I'm omitting to save space).

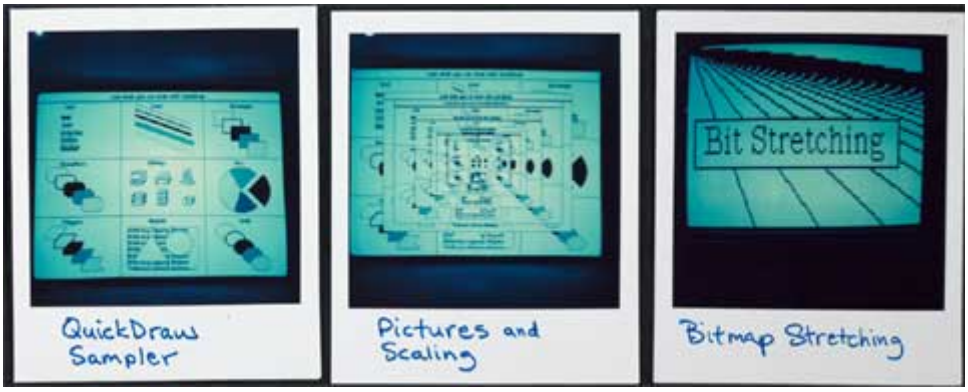



Bud Tribble was living at Bill's house now, and tended to sleep during the day and work all night, so Bill drew the phase diagram diagram on the left with the sketch program. The middle picture shows fast ovals, which were added to LisaGraf as a basic type in Spring 1981, using a clever algorithm that didn't require multiplication. They were quickly followed by rectangles with rounded corners, or "roundrects", illustrated on the right, which were suggested by Steve Jobs (see Round Rects Are Everywhere!) .

By May 1981, the Lisa user interface is beginning to solidify. The leftmost photo shows scrollable documents of different types in overlapping windows, still sporting folder tabs for titles. The middle picture shows how roundrects began to creep into various UI elements, like menus, providing a more sophisticated look, especially when combined with drop shadows. The right most photo shows how menus could be also be graphical, as well as text based.

The Lisa team was worried about the closed window tabs being obscured by other windows on the desktop, so Bill added a standard menu on the extreme left called "the tray", that could show and hide opened windows. The middle and right pictures portray a prototype that Bill created for the Lisa Graphics Editor (which eventually evolved into MacDraw), to demonstrate that modes could sometimes be useful; it was the first program to select modes with a graphical palette, which eventually became the main user interface of MacPaint.

The last major change in the Lisa User Interface was moving to an icon-based file manager in March 1982. The leftmost picture was an early mock-up done in the graphics editor, using a two-level hierarchy; selecting an icon in the top pane displays its contents in the bottom one. By the middle photo, Bill arrived at something very similar to the shipping design, complete with a trash can at the lower right. (see Rosing's Rascals) . Note that the folder tab on windows has disappeared now, replaced by a rectangular title bar that's partially inverted when highlited.




Finally, Bill renamed "LisaGraf" to "QuickDraw" in the spring of 1982, because he wanted a name that was suitable for the Macintosh, too. He added two related features to meet the burgeoning needs of the Lisa applications: pictures and scaling. Pictures were a way of recording graphics operations into a data structure for later playback; this became the basis of both our printing architecture and also cutting and pasting graphics. Since pictures could be drawn into an arbitrary sized rectangle, it also caused Bill to add bitmap scaling features as well.

---

Most users and developers only experienced the user interface as a completed whole, so they tend to think of it as static and never changing, when in fact these pictures show that it was always evolving as we gained more experience and tackled more application areas. A user interface is never good enough, and, while consistency between applications is an important virtue, the best developers will continue to innovate when faced with new problems or perhaps just when they see a much better way to accomplish something. As usual, Bob Dylan said it best when he wrote in 1965, "He not busy being born, is busy dying."

# The Macintosh Spirit

**Author:** Andy Hertzfeld
**Date:** undated
**Characters:** Steve Jobs, John Couch, Jef Raskin, Burrell Smith
**Topics:** Apple Spirit, Inspiration, Management, Personality, Origins, Reality Distortion
**Summary:** The attitudes and values of the team forged the spirit of the Macintosh
**Revision:** most recent of 17

The original Macintosh was designed by a small team that worked long hours with a passionate, almost messianic fervor, inculcated by our leader, Steve Jobs, and the excitement that we felt during its creation shines through in the finished product. The attitudes, values and personalities of the designers are reflected in the thousands of subtle choices that they make in the course of their design, coalescing into a spirit or feeling imparted to its users.

We were excited because we thought we had a chance to do something extraordinary. Most technology development is incremental, but every once in a while there's an opportunity to make a quantum leap to a whole new level. A few years earlier, the Apple II and other pioneering systems made computing affordable to individuals, but they were still much too hard for most people to use. We felt that the Mac's graphical user interface had the potential to make computing enjoyable to non-technical users for the very first time, potentially improving the lives of millions of users.

As soon as he seized the reins from Jef Raskin in January 1981, Steve Jobs galvanized the Macintosh team with an extreme sense of urgency. One of his first acts as head of the project was to bet John Couch, the executive in charge of the Lisa Division, $5000 that the Macintosh would beat the Lisa to market, despite the fact that Lisa had more than a two year head start, and we had barely begun. The Mac team always had incredibly optimistic schedules, because Steve would never be satisfied with more realistic estimates (see Reality Distortion Field) , as if he could make it happen faster through sheer force of will.

But the desire to ship quickly was counterbalanced by a demanding, comprehensive perfectionism. Most commercial projects are driven by commercial values, where the goal is to maximize profits by outperforming your competition. In contrast, the Macintosh was driven more by artistic values, oblivious to competition, where the goal was to be transcendently brilliant and insanely great. We wanted the Macintosh to be a technical and artistic tour-de-force that pushed the state of the art in every conceivable dimension. No detail was too small to matter (see PC Board Esthetics) , and good enough wasn't good enough - if Steve could perceive it, it had to be great.

Steve encouraged the Mac designers to think of ourselves as artists. In the spring of 1982, he took the entire Mac team on a field trip to a Louis Comfort Tiffany exhibition in San Francisco, because Tiffany was an artist who was able to mass produce his work, as we aspired to do. Steve even had us individually sign the interior of the Macintosh case, like artists signing their work (see Signing Party) , encouraging each one of us to feel personally responsible for the quality of the product.

Other groups at Apple had an elaborate formal product development process, mandating lengthy product requirement documents and engineering specifications before implementation commenced. In contrast, the Mac team favored a more creative, flexible, incremental approach of successively refining prototypes. Burrell Smith developed a unique hardware design style based on programmable array logic chips (PAL chips), which enabled him to make changes much faster than traditional techniques allowed, almost with the fluidity of software. Instead of arguing about new software ideas, we actually tried them out by writing quick prototypes, keeping the ideas that worked best and discarding the others (see Busy Being Born) . We always had something running that represented our best thinking at the time.

You might think that impossible schedules and uncompromising perfectionism would lead to an oppressive work environment, but most of the time, the ambiance of the Mac team was spontaneous, enthusiastic and irreverent. Jef Raskin had a playful management style, encouraging a workplace teeming with toys and semi-organized games (see Good Earth) , which carried over to the Jobs era. Most of the early team members were around the same age, in our mid-twenties, and we enjoyed each other's company. We increasingly hung out together as the project demanded ever greater chunks of our time, abandoning the distinction between work and play. Despite the incessant pressure, we loved what we were doing.

Given Steve's autocratic tendencies, the Mac team was surprisingly egalitarian. Unlike other parts of Apple, which were becoming more conservative and bureaucratic as the company grew, the early Mac team was organized more like a start-up company. We eschewed formal structure and hierarchy, in favor of a flat meritocracy with minimal managerial oversight, like the band of revolutionaries we aspired to be. Steve Jobs would sometimes issue an unreasonable edict or veto something that everyone else wanted, but at least he would relent when he saw he was wrong (see Quick, Hide In This Closet!) . At our third retreat in January 1983, Steve reinforced our rebel spirit, which was waning as the team grew larger, by telling us "it's better to be a pirate than join the navy" (see Pirate Flag) .

Enthusiasm is contagious, and a product that is fun to create is much more likely to be fun to use. The urgency, ambition, passion for excellence, artistic pride and irreverent humor of the original Macintosh team infused the product and energized a generation of developers and customers with the Macintosh spirit, which continues to inspire more than twenty years later.

# The Apple Spirit

**Author:** Andy Hertzfeld
**Date:** undated
**Characters:** Andy Hertzfeld
**Topics:** Apple Spirit, Inspiration
**Summary:** An essay I wrote about Apple's values that was published in MacWeek
**Revision:** most recent of 1

*Toward the end of 1988, I wrote an essay that was published in MacWeek entitled "The Apple Spirit". It was about the creative magic that I found in the Apple II, and how we were able to transplant it into the Macintosh. The essay articulates the values behind lots of the stories collected here, so I thought it was worth including.*

**The Apple Spirit**
November 29, 1988

The best purchase of my life occurred in January 1978 when I spent $1295 plus tax (most of my life savings at the time) on an Apple II microcomputer (serial number 1703) with 16K bytes of RAM. I was instantly delighted with it, and the deeper I dug into it, the more excited I became. Not only could I finally afford to have my own computer, but the one I got turned out to be magic; it was better than I ever thought it possibly could be!

I started spending most of my free time with my Apple, and then most of my not-so-free time, exploring the various technical aspects of the system. As I taught myself 6502 assembly language from the monitor listing that came with the machine, it became clear to me that this was no ordinary product; the coding style was crazy, whimsical and outrageous, like every other part of the design, especially the hi-res graphic screen; it was clearly the work of a passionate artist. Eventually, I became so obsessed with the Apple II that I had to go to work at the place that created it. I abandoned graduate school and started work as a systems programmer at Apple in August 1979.

Even though the Apple II was overflowing with both technical and marketing genius, the best thing about it was the spirit of its creation. It was not conceived or designed as a product in the usual sense; it was just Steve Wozniak trying to impress himself and his friends. Most of the early Apple employees were their own ideal customers. The Apple II was simultaneously a work of art and the fulfillment of a dream, shared by Apple's employees and customers. Its unique spirit was picked up and echoed back by third party developers, who sprung out of nowhere with innovative applications.

The personal computer industry began to grow and evolve very rapidly when larger companies realized the extraordinary potential of personal computers. Apple's sales took off like a skyrocket as the Apple II became accepted as an established industry standard. By the time the early 1980s rolled around, many opportunists had come to both Apple and the personal computer industry, people whose only concern was to make as much money as possible. I started to become disillusioned when Apple hired many professional managers who didn't appreciate the magic of the Apple II; many of them would have been just as happy selling refrigerators. I probably would have left Apple sometime in 1981 if I hadn't run across a tiny, sloppily wire-wrapped digital board created by Burrell Smith, a young technician who worked in the service department.

Burrell worshipped Woz's Apple II design and had forged an idiosyncratic design style that was even crazier than Woz's, using many clever tricks to coax enormous functionality out of the minimum number of chips. Somehow, Burrell's embryonic Macintosh board reeked of the same creative spirit so prevalent in the Apple II; as soon as I saw it, I knew that I had to work on the project.

Steve Jobs also became enamored with Burrell's circuit board and quickly took over the tiny design group, moving it to a remote part of the company and inspiring us with a grand vision. The Apple II had broken through an important price barrier, making a useful personal computer affordable to ordinary individuals, but it was still much to hard for most non-technical people to master. The Macintosh would harness the potential of Motorola's 68000 microprocessor to become the first personal computer that was both easy to use and affordable. We thought that we had a chance to create a product that could make computers useful to ordinary people and thereby truly change the world.

The Macintosh design team was inspired by Woz's original design and tried to recapitulate its innovative spirit. Again, we were our own ideal customers, designing something that we wanted for ourselves more than anything else. Although Apple was already a large company by then, Steve's unique position in the organization enabled him to maintain the Macintosh group as a little island where Apple's original values could flourish and grow. The Macintosh was released in January 1984 and eventually became a very successful product.

The personal computer industry has continued to grow and change since the introduction of the original Macintosh. Apple has become [in 1988] a four billion dollar enterprise, and I often fear that they have lost touch with their original values. Yet I remember having similar worries right before starting work on the Mac. I'm sure that there are little groups at Apple right now, inspired by the Macintosh in exactly the same way that we were inspired by the Apple II. The great challenge facing Apple's management is to allow those groups to follow their hearts and imaginations, uncompromised by the inevitable politics of large organizations. I hope that I will be able to buy a new Apple computer in 1991 that is not a Macintosh or an Apple II, but rather an entirely new system that once again shares the maverick spirit of its illustrious ancestors.

# The Father Of The Macintosh

**Author:** Andy Hertzfeld

**Date:** undated

**Characters:** Bud Tribble, Steve Jobs, Joanna Hoffman, Jef Raskin, Rod Holt, Burrell Smith, Brian Howard, Steve Wozniak, Jerry Manock, Bill Atkinson, Mike Markkula

**Topics:** Credit, Personality

**Summary:** Who is the father of the Macintosh?

**Revision:** most recent of 27

In the early days of the personal computer industry, breakthrough products could still be created single-handedly, or by very small teams. Steve Wozniak is indisputably the father of the Apple II, having designed the entire digital board himself as well as writing all of the system software, including a BASIC interpreter, most of it before Apple was even incorporated. But even Woz required help from Rod Holt for the analog electronics (the Apple II's switching power supply was almost as innovative as the digital board) and Steve Jobs and Jerry Manock for the industrial design (ditto for the plastic case).

By the 1980s, things had gotten more complicated. The Macintosh was more of a team effort, with at least a half dozen people making significant, invaluable contributions. For the launch publicity, Steve Jobs anointed seven of us (not counting himself) as the official "design team", but it could just as easily have been five or fifteen. Some people felt bad that they weren't included, and it was obvious that there was no good way to draw the line.

But if you look up the phrase "Father of the Macintosh" on Google, you get lots of links mentioning the initiator of the project, Jef Raskin. Jef was a former professor at UCSD (of both computer science and music) who started at Apple in January 1978 as Apple employee #31, after contracting to write Apple's manual for Basic with his friend Brian Howard, at their consulting firm named Bannister and Crun (a playful name appropriated from the Goon Show). Apple liked the Basic manual so much that they hired Jef and Brian to be founders of their internal publications group.

**Jef Raskin**

In early 1979, after successfully building an outstanding pubs department, Jef turned the reins over to Phyllis Cole and started thinking about what it would take for personal computers to expand beyond the current hobbyist market, writing up his ideas in a series of short papers. He presented his idea for an ultra low cost, easy to use appliance computer to Mike Markkula in March 1979, and got the go-ahead to hire a few people and form an official research project later in September 1979, naming it Macintosh, after his favorite kind of eating apple. Most of his ideas for the new machine were collected in a set of papers he called "The Book of Macintosh".

There's no doubt that Jef was the creator of the Macintosh project at Apple, and that his articulate vision of an exceptionally easy to use, low cost, high volume appliance computer got the ball rolling, and remained near the heart of the project long after Jef left the company. He also deserves ample credit for putting together the extraordinary initial team that created the computer, recruiting former student Bill Atkinson to Apple and then hiring amazing individuals like Burrell Smith, Bud Tribble, Joanna Hoffman and Brian Howard for the Macintosh team. But there is also no escaping the fact that the Macintosh that we know and love is very different than the computer that Jef wanted to build, so much so that he is much more like an eccentric great uncle than the Macintosh's father.

Jef did not want to incorporate what became the two most definitive aspects of Macintosh technology - the Motorola 68000 microprocessor and the mouse pointing device. Jef preferred the 6809, a cheaper but weaker processor which only had 16 bits of address space and would have been obsolete in just a year or two, since it couldn't address more than 64Kbytes. He was dead set against the mouse as well, preferring dedicated meta-keys to do the pointing. He became increasingly alienated from the team, eventually leaving entirely in the summer of 1981, when we were still just getting started, and the final product utilitized very few of the ideas in the Book of Macintosh. In fact, if the name of the project had changed after Steve took over in January 1981, and it almost did (see Bicycle) , there wouldn't be much reason to correlate it with his ideas at all.

So, if not Jef, does anyone else qualify as a parent of the Macintosh? Bill Atkinson is a strong candidate, since he was almost singlehandedly responsible for the breakthrough user interface, graphics software and killer application that distinguished the Mac. A case could also be made for Burrell Smith, whose wildly creative digital board was the seed crystal of brilliance that everything else coalesced around. But ultimately, if any single individual deserves the honor, I would have to cast my vote for the obvious choice, Steve Jobs, because the Macintosh never would have happened without him, in anything like the form it did. Other individuals are responsible for the actual creative work, but Steve's vision, passion for excellence and sheer strength of will, not to mention his awesome powers of persuasion, drove the team to meet or exceed the impossible standards that we set for ourselves. Steve already gets a lot of credit for being the driving force behind the Macintosh, but in my opinion, it's very well deserved.

# On Xerox, Apple and Progress

**Author:** Bruce Horn
**Date:** undated
**Characters:** Bruce Horn, Bill Atkinson, Larry Tesler, Steve Jobs, Jerry Manock
**Topics:** Credit, Software Design, Origins
**Summary:** An essay discussing where the Mac user interface came from
**Revision:** most recent of 4

This essay was written by Bruce in 1996, and is reprinted here with his permission. Bruce was one of the main designers of the Macintosh software, and he worked at Xerox for years before that, so he's uniquely qualified to discuss their relationship.

---

### Where It All Began

For more than a decade now, I've listened to the debate about where the Macintosh user interface came from. Most people assume it came directly from Xerox, after Steve Jobs went to visit Xerox PARC (Palo Alto Research Center). This "fact" is reported over and over, by people who don't know better (and also by people who should!). Unfortunately, it just isn't true - there are some similarities between the Apple interface and the various interfaces on Xerox systems, but the differences are substantial.

Steve did see Smalltalk when he visited PARC. He saw the Smalltalk integrated programming environment, with the mouse selecting text, pop-up menus, windows, and so on. The Lisa group at Apple built a system based on their own ideas combined with what they could remember from the Smalltalk demo, and the Mac folks built yet another system. There is a significant difference between using the Mac and Smalltalk.

Smalltalk has no Finder, and no need for one, really. Drag-and- drop file manipulation came from the Mac group, along with many other unique concepts: resources and dual-fork files for storing layout and international information apart from code; definition procedures; drag-and-drop system extension and configuration; types and creators for files; direct manipulation editing of document, disk, and application names; redundant typed data for the clipboard; multiple views of the file system; desk accessories; and control panels, among others. The Lisa group invented some fundamental concepts as well: pull down menus, the imaging and windowing models based on QuickDraw, the clipboard, and cleanly internationalizable software.

Smalltalk had a three-button mouse and pop-up menus, in contrast to the Mac's menu bar and one-button mouse. Smalltalk didn't even have self-repairing windows - you had to click in them to get them to repaint, and programs couldn't draw into partially obscured windows. Bill Atkinson did not know this, so he invented regions as the basis of QuickDraw and the Window Manager so that he could quickly draw in covered windows and repaint portions of windows brought to the front. One Macintosh feature identical to a Smalltalk feature is selection-based modeless text editing with cut and paste, which was created by Larry Tesler for his Gypsy editor at PARC.

As you may be gathering, the difference between the Xerox system architectures and Macintosh architecture is huge; much bigger than the difference between the Mac and Windows. It's not surprising, since Microsoft saw quite a bit of the Macintosh design (API's,sample code, etc.) during the Mac's development from 1981 to 1984; the intention was to help them write applications for the Mac, and it also gave their system designers a template from which to design Windows. In contrast, the Mac and Lisa designers had to invent their own architectures. Of course, there were some ex- Xerox people in the Lisa and Mac groups, but the design point for these machines was so different that we didn't leverage our knowledge of the Xerox systems as much as some people think.

The hardware itself was an amazing step forward as well. It offered an all-in-one design, four-voice sound, small footprint, clock, auto-eject floppies, serial ports, and so on. The small, portable, appealing case was a serious departure from the ugly- box-on-an-ugly-box PC world, thanks to Jerry Manock and his crew. Even the packaging showed amazing creativity and passion - do any of you remember unpacking an original 128K Mac? The Mac, the unpacking instructions, the profusely-illustrated and beautifully- written manuals, and the animated practice program with audio cassette were tastefully packaged in a cardboard box with Picasso- style graphics on the side.

**Looking Back**

In my opinion, the software architectures developed at Xerox for Smalltalk and the Xerox Star were significantly more advanced than either the Mac or Windows. The Star was a tremendous accomplishment, with features that current systems haven't even started to implement, though I see OpenDoc as a strong advance past the Xerox systems. I have great respect for the amazing computer scientists at Xerox PARC, who led the way with innovations we all take for granted now, and from whom I learned a tremendous amount about software design.

Apple could have developed a more complex, sophisticated system rivaling the Xerox architectures. But the Mac had to ship, and it had to be relatively inexpensive - we couldn't afford the time or expense of the "best possible" design. As a "little brother" to the Lisa, the Macintosh didn't have multitasking or protection - we didn't have space for the extra code or stack required. The original Macintosh had extremely tight memory and disk constraints; for example, the Resource Manager took up less than 3,000 bytes of code in the ROM, and the Finder was only 46K on disk. We made _many_ design decisions that we regretted to some extent - even at the time some of us felt disappointed at the compromises we had to make - but if we had done it differently, would we have shipped at all?

**The Past and Future**

In many ways, the computing world has made remarkably small advances since 1976, and we continually reinvent the wheel. Smalltalk had a nice bytecoded multi-platform virtual machine long before Java. Object oriented programming is the hot thing now, and it's almost 30 years old (see the Simula-67 language). Environments have not progressed much either: I feel the Smalltalk environments from the late 1970's are the most pleasant, cleanest, fastest, and smoothest programming environments I have ever used. Although CodeWarrior is reasonably good for C++ development, I haven't seen anything that compares favorably to the Smalltalk systems I used almost 20 years ago. The Smalltalk systems of today aren't as clean, easy to use, or well-designed as the originals, in my opinion.

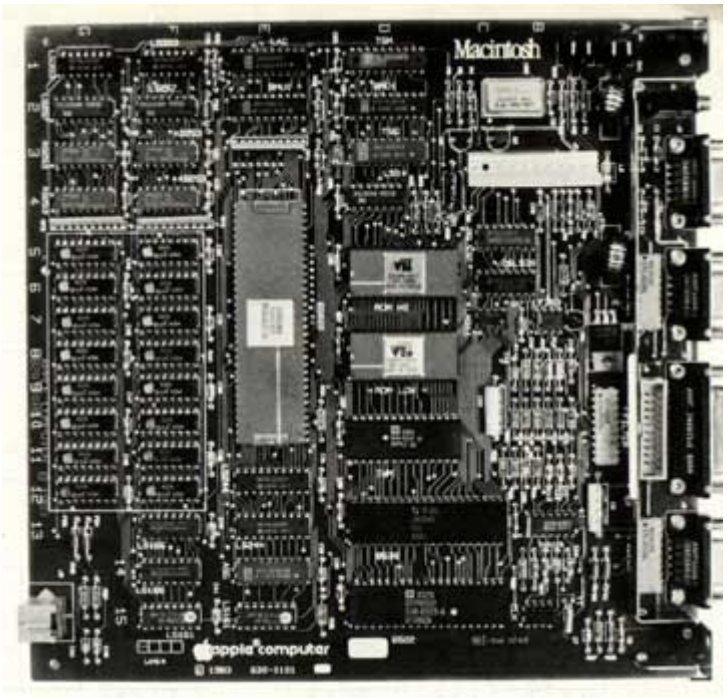
We are not even _close_ to the ultimate computing-information- communication device. We have much more work to do on system architectures and user interfaces. In particular, user interface design must be driven by deep architectural issues and not just new graphical appearances; interfaces are structure, not image. Neither Copland nor Windows 95 (nor NT, for that matter) represent the last word on operating systems.

Unfortunately, market forces are slowing the development of the next revolution. Still, I think you can count on Apple being the company bringing these improvements to next generation systems.

I'm sure some things I remember as having originated at Apple were independently developed elsewhere. But the Mac brought them to the world.

# Five Different Macintoshes

**Author:** Andy Hertzfeld
**Date:** undated
**Characters:** Burrell Smith, Bud Tribble, Wendell Sander, Steve Jobs, Jef Raskin, Bill Atkinson
**Topics:** Hardware Design, Technical, Prototypes
**Summary:** Burrell actually designed 5 different Macintoshes
**Revision:** most recent of 7




**The final version of the Mac digital board**

The awesomely creative design of the Macintosh digital board was always the seed crystal of brilliance at the core of the project, but there wasn't just one design; Burrell redesigned the digital board four different times as development proceeded, finally arriving at the shipping design in the fall of 1982.

The first Macintosh digital board, designed in late 1979, was based on Jef Raskin's specifications: it had a Motorola 6809E microprocessor, 64K of memory, and a 256x256 black and white, bit-mapped graphics display. Even in this first design, Burrell was using his trademark 'PAL' chips, which were small, programmable logic arrays, to provide all of the system glue, so the system had a very low chip count.

The first Macintosh was a cute little computer, but it was fundamentally limited by the 6809 microprocessor, which only had 16 bits of address space. Bill Atkinson was doing incredible work on the Lisa project using Motorola's 68000 microprocessor, with its capacious 32 bit registers and 24 bit address space. Bud Tribble, the Mac's only software person, was living at Bill's house, and watching the extraordinary progress on Lisa's graphics package. He began to wonder if it was possible for the Macintosh to use the 68000, so it could run Bill's graphics routines.

Bud began asking Burrell if it was feasible to include the 68000 in a low cost design. The 68000 was expensive enough on its own, but its 16 bit memory bus required twice as many RAM chips as the 6809, so the overall cost was significantly more expensive. But Burrell thought about the problem and came up with a characteristically brilliant idea for his second Macintosh design.

The idea was what Burrell called a "bus transformer" circuit, built out of PAL chips, which adapted the 68000 to an 8 bit memory bus by exploiting the fast "page mode" access mode of the RAMs. The new Macintosh, designed over the Christmas break at the end of 1980, featured an 8 megahertz 68000, 64K of RAM, and a 384 by 256 bit mapped display. It was 60% faster than the Lisa (which used a 5 megahertz 68000) but a lot less expensive.

When Steve Jobs caught wind of what Burrell had come up with - an Apple II priced machine that blew away the Lisa, he became really excited, and saw that Burrell's machine could become the future of Apple. Steve's attention was the beginning of the end for Jef, though, who despised parts of Steve's personality and couldn't put up with Steve's courting of Burrell and Bud. Steve took over the project in January 1981, and the Macintosh entered the post-Jef era, on track to becoming a real product.

Burrell's third Macintosh design was done in June 1981. The main reason was that he fell in love with a communications chip called the SCC. The SCC could support a built-in local area network, making AppleTalk possible with no additional hardware, as well as providing nice buffered serial ports with interrupts and other hardware features. At this point it also was becoming obvious that we needed at least 128K of memory to support the user interface, so he also added a second row of RAM chips.

Around the end of 1981, Burrell met some engineers who were doing custom LSI chips, which were very flexible and powerful but very time consuming to design, since the software tools were still in their infancy. Burrell decided that he wanted to take a shot at building the Macintosh around a single custom chip, and convinced Steve that he could pull it off. Instead of sitting around doing nothing while the software team finished the product, he would work with a couple of experienced LSI designers and redesign the Mac around what he called the "Integrated Burrell Machine".

Earlier in the year, Wendell Sander, the designer of the Apple III and one of Apple's best engineers, did a small custom chip that crammed all the functionality of Woz's disk controller into a single chip. It was called the "IWM" chip, which stood for the "Integrated Woz Machine", since Woz's disk controller is really an elaborate state machine, but it also stood for the "Integrated Wendell Machine". So when Burrell envisioned a single custom chip for the Macintosh, he called it the "IBM" chip, for "Integrated Burrell Machine", relishing the potential confusion the name might cause.

In the beginning of 1982, the original 68000 design was more than a year old, and the software was nowhere near finished, so Burrell was afraid some of the trade-offs of the original design were no longer current. He used the expansive canvas of a custom chip, where additional logic was almost free, to update the architecture. The most important decision was admitting that the software would never fit into 64K of memory and going with a full 16-bit memory bus, requiring 16 RAM chips instead of 8. The extra memory bandwidth allowed him to double the display resolution, going to dimensions of 512 by 342 instead of 384 by 256. He also added bells and whistles like a fancy, DMA-fed sound generator with four independent voices. This was the fourth version of the Macintosh design.

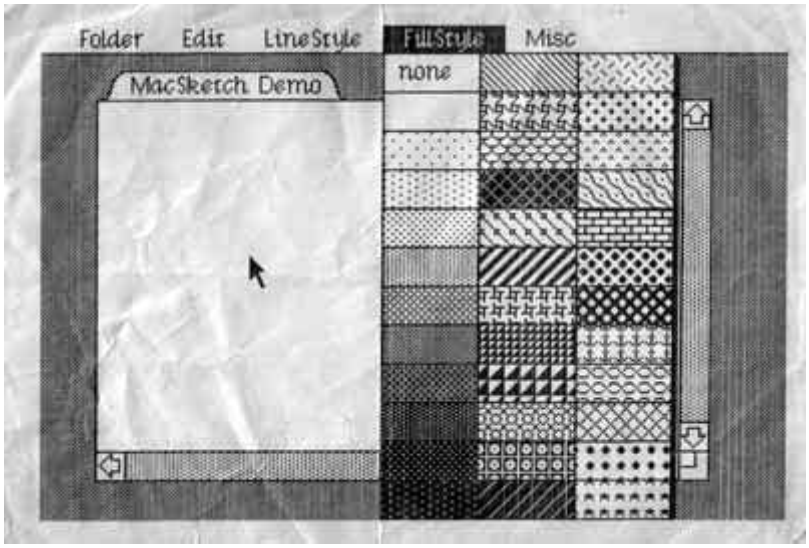
But Burrell had never done a custom chip before. He was counting on experts working with our partner VLSI Design, mainly Doug Fairbairn and Martin Haeberli, who successfully made a custom chip for an optical mouse, but they had never really worked against commercial deadlines, and a lot of tension began to build as the schedule started to slip.

Some more of the details of the Integrated Burrell Machine project are told elsewhere, but eventually, after the first silicon from an early version didn't work as expected, Burrell became frustrated and decided that it was too risky to continue. He realized that he could do a design with PALs that achieved most of the gains from the Integrated Burrell Machine, without needing the custom chip. So, in August 1982, he quickly designed the fifth iteration of the Macintosh, the one that actually shipped in January 1984, by adapting his earlier PAL-based design to one that had a 16-bit memory bus with a 512 by 342 display and some other features invented for the custom chip.

# Busy Being Born, Part 2

**Author:** Andy Hertzfeld
**Date:** undated
**Characters:** Bud Tribble, Bill Atkinson, Bruce Horn, Steve Capps, Susan Kare
**Summary:** Here are a few seminal Macintosh screenshots, ala the Lisa polaroids
**Revision:** most recent of 22

Bill Atkinson had the foresight to document the creation of the Lisa User Interface by keeping a polaroid camera near his computer, taking a snapshot of each significant milestone (see Busy Being Born) . Although we didn't systematically save pictures of key Mac milestones, I've managed to cobble together a few seminal Macintosh screenshots to present here in a similar fashion.




**the first Mac-like demo in May 1981**

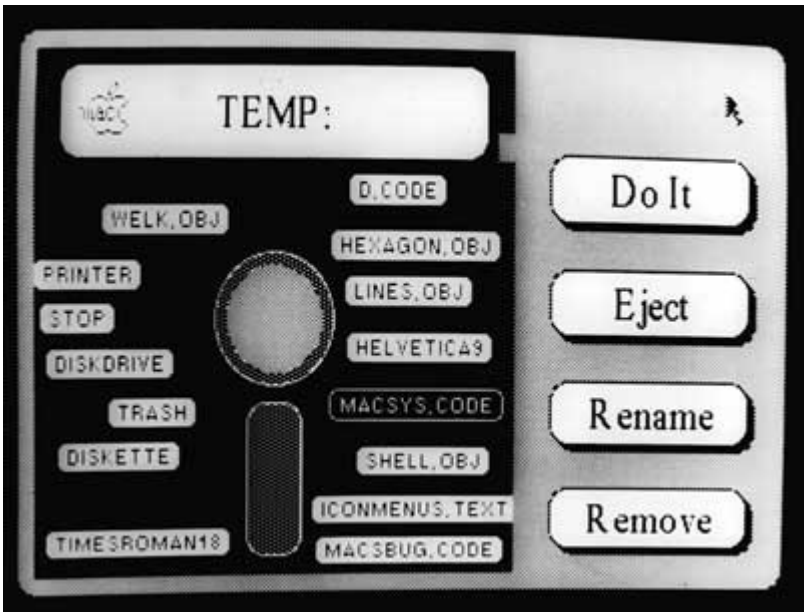
Bud Tribble had a tendency to work late at night. I usually came to work at Texaco Towers around 10:30AM, so if Bud was there when I arrived, it usually meant that he had spent all night there. One morning, in the middle of May 1981, when I arrived at my usual time, Bud was anxious to show me something before I could even take off my backpack.

I knew that Bud had been working on the initial porting of QuickDraw to the Macintosh, but I thought that he was at least a week away from getting it running. At this point, we had some cursor routines going, and a way to download and execute Pascal programs that were compiled on a Lisa, attached to the Mac by a serial cable. But we didn't have a memory manager yet, or an event manager or file system, so Bud had to build scaffolding in various places to overcome these limitations. He had compiled a bitmap drawing program that Bill wrote in Pascal for Lisa, then linked it with LisaGraf and other library routines, and started to debug it, fixing each problem as it manifested.

Bud had made a huge amount of progress the previous evening, and the demo was substantially running now. It was incredibly exciting to see Mac-like software running on the Mac for the very first time. The demo featured working pull-down menus, complete with a nicer style of drop shadow than the Lisa was using, and an elaborate, graphical pattern menu, which is illustrated in the screenshot above.

Xerox aficionados will note the use of Cream 12 as our first system font, which was the default font used by SmallTalk, that Bill had converted to the Lisagraf font format. The window title bar was a folder tab, because we were still confused about the difference between folders and documents. The demo already had scroll bars and a grow box that is pretty similar to what we ended up shipping with, although you couldn't interact with them yet. In fact, the only part of the program that actually did something was the "Quit" command.

In April, I had written some screen printing code, that dumped whatever was on the display out the serial port to a dot matrix printer. Since the Mac screen was rather small, I added a feature to print it at double size, so it mostly filled a page. I used that to print the display of Bud's demo, with the impressive graphical pattern menu pulled down, on the very day that Bud got it working, and that's what is reproduced above.

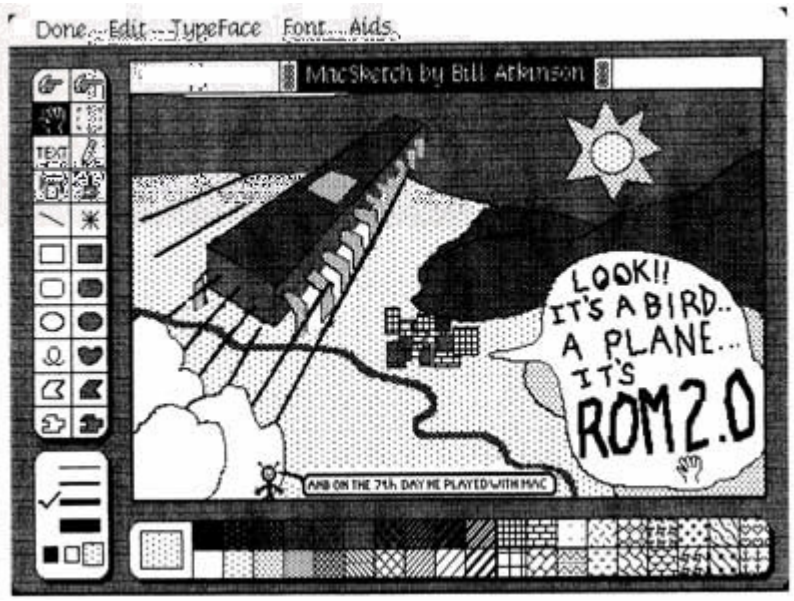



**An alternate approach to the Finder in March 1982**

Bruce Horn joined the Mac team in late 1981 (see Joining the Mac Group) , with the charter to write a graphical shell that we were calling the "Finder", since it helped the user find applications and documents to launch. We were influenced by ideas from the Architecture Machine group at MIT (a predecessor to the better known Media Lab) as portrayed in a program called "DataLand" that allowed users to manipulate graphical objects in spacial arrangements. Bruce was excited about spacial data management and his first assignment was to write a prototype to explore how it could work on the Mac.

Bruce came up with the idea of representing files as small tabs superimposed on an image of a floppy disk. He wrote a prototype that he called "the micro-finder", which is pictured above. I started helping him implement various parts of it, and pretty soon it was actually useful. You could drag the file tabs to position them, and click on the large buttons on the right to launch programs or rename and delete files. We used the micro-finder through most of 1982 for demoing the Mac until the real Finder started becoming usable around the end of the year.

After the micro-finder, Bruce also worked on another prototype that included folders in a two-pane view (Bruce thinks he can eventually dig up a picture of it; if he does I'll include it here). Meanwhile, Bill Atkinson was crafting an icon-based file manager prototype for Lisa (see Rosing's Rascals) , and eventually we decided to follow that direction for the Macintosh.




**An early screenshot of a half-implemented MacPaint**

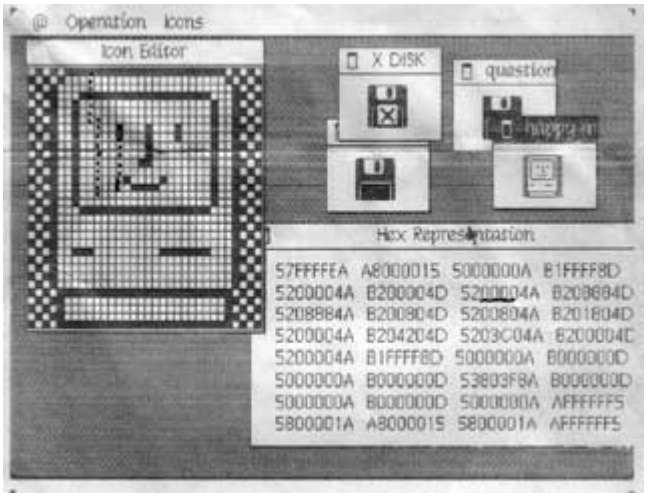
Here is a very early version of MacPaint, probably from March 1983, after Bill had been working on it for around one month. The first thing to notice is that it wasn't called "MacPaint" yet - it still bore its original name, "MacSketch", inherited from its predecessor, LisaSketch.

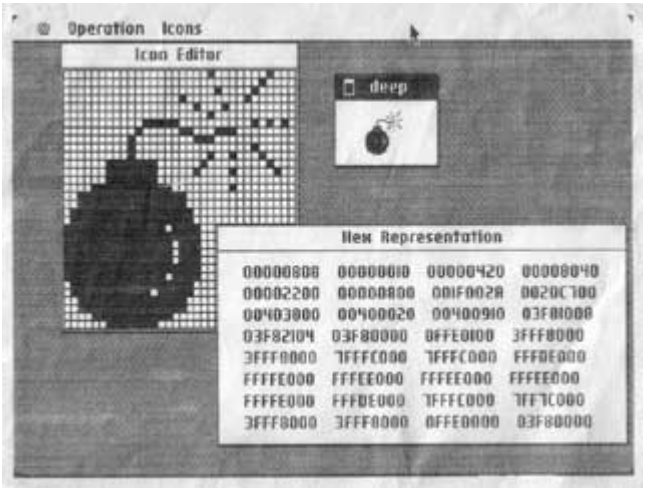
This early version uses icons designed by Bill himself, before Susan Kare got a chance to tweak them. And some of the most important MacPaint tools like the paint bucket and the lasso are still months away from being implemented.

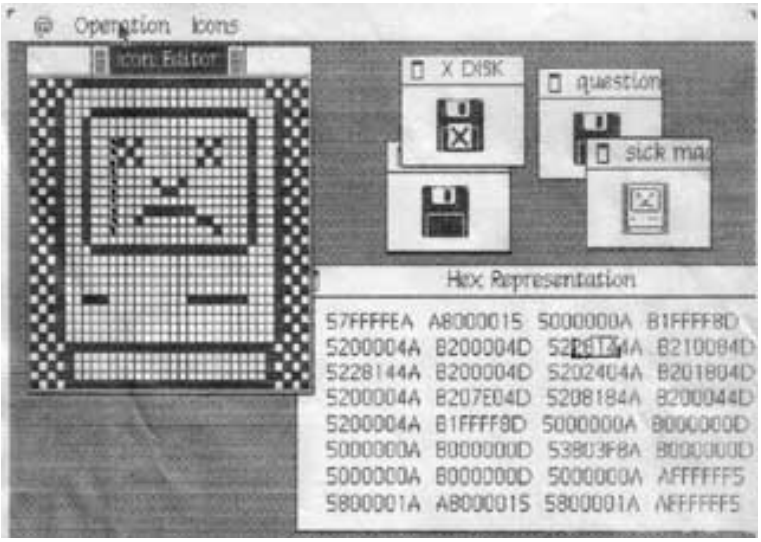
MacPaint contained a menu of miscellaneous tools, like Fat Bits, originally called the "Aids" menu, as you can see above. But in the summer of 1983, with public awareness of the AIDS epidemic beginning to swell, Bill rechristened it the "Goodies" menu.

It's interesting to note the window highlighting decorations, which are quite different than what we ended up with. We must have tried dozens of different ways to highlight windows before arriving at the horizontal lines in August 1983.

The featured MacPaint document was drawn late one night by Steve Capps, to celebrate one of our ROM releases; he also saved and scanned the document for inclusion here.





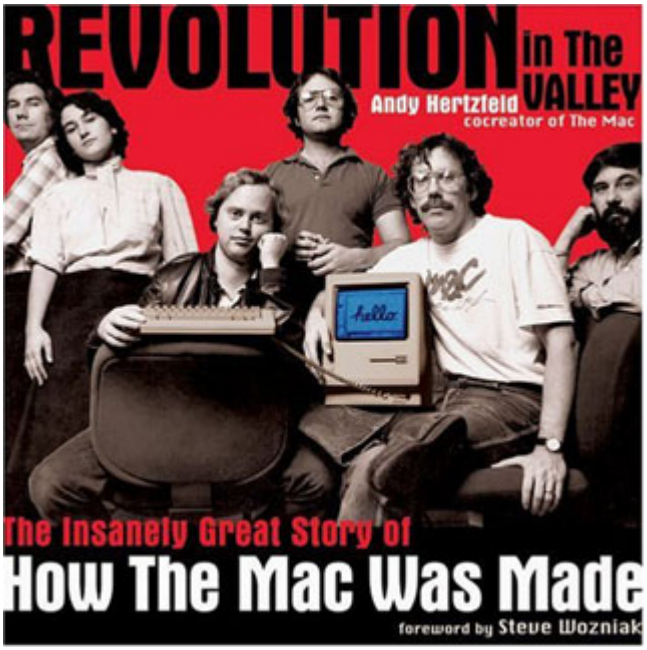

In early 1983, I wrote an icon editor based on Bill Atkinson's "Fat Bits" pixel editing techniques that Susan Kare used to craft most of the early Mac icons. The icon editor displayed both a large and actual size representation of the icon, and allowed editing multiple icons at once. I needed a way to incorporate the icons into the ROM, so I added a feature called the "Hex Window", that displayed the representation of the current icon in hexadecimal, which is what I needed to add the icons to the Mac ROM source code.

The screen dumps on the left are the actual ones used to enter some famous Mac icons into ROM, like the bomb and the happy Mac.

Note that the name on the window containing the bomb icon is "Deep". That's the first word of the original name of the code that displayed the dreaded bomb icon on the screen. Originally the code had an obscene name, with the API calls prefixed with "DS". Jerome came up with various euphemisms for it, like the "Deep Sauce" manager evolving into the "Dire Straights" manager, eventually settling on the more prosaic "System Error" manager.

# Revolution in the Valley

**Author:** Andy Hertzfeld
**Date:** undated
**Characters:** Steve Jobs, Andy Hertzfeld, Mike Markkula, Steve Wozniak, Tim O'Reilly, Allen Noren, Michelle Weatherbee, Melanie Wang
**Topics:** Commentary, The Book
**Summary:** The story of how this site turned into a book
**Revision:** most recent of 34




**Revolution in the Valley** is a new hardcover book, published by O'Reilly Media in December 2004, that is based on the anecdotes from this website. This story describes the book and how it came to be, and what it means to Folklore.org moving forward.

When I began to write stories for this site in June 2003, I had no intention of trying to publish them in book form. I was excited by the idea of developing a web site to facilitate collective historical storytelling, where a group of participants could collaborate to recollect an interesting event. I chose the format of numerous interlinked anecdotes because it seemed natural for the web and better suited to a collaborative effort than a single, continuous narrative, allowing a tale to be elaborated indefinitely without compromising the voices of the individual authors.

After I got the initial site running in August 2003, with about 20 stories, I began to show it to various original Mac team members and others, to gather feedback and encourage participation. When I showed the site to Tim O'Reilly, I was surprised he suggested that his company publish it as a book. At first, I thought conforming to a book format might compromise my goals for the site, but I soon realized that the site's anecdotal structure could work in book form and I got excited about the idea. After all, I own dozens of indispensable O'Reilly books, so I was thrilled at the chance to become one of their authors. Without showing it to other publishers, I signed a contract with O'Reilly in December 2003, promising a finished draft by June 2004.

Tim introduced me to the talented team at his company, including my editor, Allen Noren, who specializes in O'Reilly's more humanistic efforts (his previous two books were Dan Gillmor's "We the People" and Paul Graham's "Hackers and Painters"). Allen guided me through the laborious process to transform the raw material of the Folklore site into a beautiful book.

After completing the writing phase in June 2004, we embarked on the editing and layout process. The first step was copy editing. All 90 or so of the stories had to be thoroughly edited, in batches of 10 at a time. Most of the changes involved fixing grammatical errors and punctuation, and removing unnecessary verbiage, chopping up my Proustian run-on sentences (like this one), but they occasionally involved additional writing to provide more explanation or clarification. My editors also wanted to delete many of the most technical passages, fearing they would alienate

non-technical readers. After editing was completed on a batch, I had to accept or reject each individual change. I accepted all the grammatically oriented ones, figuring my editors knew better than I did, but I fought to keep most of the technical detail, since I thought it was an important part of the story, although we did eliminate a few of the most technical stories entirely. Sometimes we'd argue about the merits of particular changes, but it usually wasn't that hard to reach consensus. I decided to keep the original text on the site, so you can compare it with the book if you're interested in seeing what changed.

The next step was working on the layout. We decided to maintain the basic format of the site, but we divided the stories into five parts at the natural breaking points, to give the reader a chance to rest. We decided to use an unusual form factor, eight inches square, which seems simultaneously large and small at the same time. I was delighted when Allen told me they wanted to print it in full color on high quality paper without increasing the price.

Unfortunately, I discovered that it's a lot harder to publish images in a book than on the web. On the Folklore site, I didn't have to worry too much about obtaining permissions, since if anybody ever complained, I could easily remove an offending image. But once published, a book is immutable, and my contract required that I obtain formal rights for every image in the book. Some photographs require multiple layers of permissions (from the subjects, the photographer and the original publication); with twenty year old photographs it's difficult to track everything down. That kind of work is not my forte, but luckily Allen introduced me to a consultant who was willing to obtain permissions for $50 per image. To make matters worse, the printed page still has much higher resolution than a computer display, so we needed to procure higher resolution images than the ones on the site.

Before unveiling the site publicly in January 2004, I gathered up my courage and showed it to Steve Jobs. He was fairly enthusiastic about it, but as usual he had some complaints. "I like what you've done with the site", he told me, "but the quality of some of the scans you're using is terrible! Can't you do better than that?" When I complained that it was hard to find pristine copies of decades old material, he suggested that I could probably access Apple's corporate marketing archive to find better versions of lots of the images.

I didn't even know that Apple had a marketing archive. It took a few months to track down the right people, since I didn't want to bother Steve about the details, but I eventually found Sue Runfola who works on rights and permissions in Apple's Legal Department, who introduced me to Del Smith Penny, who maintains Apple's marketing archive as a part-time job. The archive is just a single room in a non-descript building a few blocks away from the main Apple campus, stuffed to the gills with maybe 40 long file cabinets. There were stacks of cardboard boxes on top of the cabinets that Del told me were acquired from Mike Markkula's garage, containing marketing material from Apple's first four years that no one had time to examine yet.

Some of the material in the archive was indexed in a Filemaker database running on an old Mac, but Del admitted that much of it still wasn't indexed at all, since he barely had time to keep up with the new stuff coming in. But he was willing to help me search for everything I asked him about and by the end of the afternoon Del and I were able to locate a treasure trove of around 30 high quality slides of relevant product and publicity photos, including some that were never published before. We also uncovered a rare video tape of potential TV commercials shot in October 1983 featuring the Mac design team that were never aired, but that's another story.

Allen introduced me to Michelle Weatherbee, an award winning book designer who had just hired on full-time to O'Reilly as art director; my book was her first project as an O'Reilly employee. Michelle had me bring up lots of my old Macintosh relics to O'Reilly headquarters in Sebastopol, including my design notebooks, which she borrowed for a few weeks to scan.

I worked with Michelle and layout artist Melanie Wang to match the images to the proper stories. Michelle and a few others at O'Reilly helped choose other relevant images from commercial image clearinghouses like Corbis.

Allen told me that I needed a foreword for the book and suggested that I ask Steve Jobs to write one, but I didn't have the courage to ask him to do something like that. I suggested Apple co-founder Steve Wozniak instead, who told me that he loved the site and had even contributed a few comments to some of the stories. Woz was enthusiastic about writing the foreword, which he thought he could get done in a few weeks. We told him we needed it by the end of June.

I warned Allen that while Woz always had the best intentions, he often was a notorious procrastinator. If he didn't do something right away, it had a tendency not to get done indefinitely. Predictably, the foreword wasn't finished by the end of July and Allen began to send Woz emails to remind him about it every few weeks. Woz always replied promptly, promising to make room in his busy schedule to get it done soon.

Finally, in September, while I was meeting with Allen and team in Sebastopol to finalize the layout, he told me that he had some bad news. He had finally given up on Woz's foreword, because Woz hadn't responded to an email informing him the final deadline was approaching, even though he sent it a few days ago. I laughed and told Allen that was actually good news, because not responding to the email meant that Woz was probably actually writing it, and sure enough I was right. A few days later Woz sent Allen an eloquent, stirring foreword that was a perfect start for the book.

I assumed the title of the book would be "Macintosh Folklore", just like the website, but while I was working with Michelle and Melanie on the layout, I noticed them referring to the book as "Revolutionaries". I asked Allen what was up and he told me that the sales department didn't think my title was appealing enough, and sales honcho Mark Brokering had renamed the book "Revolutionaries in the Valley", but they were waiting to tell me until they had the cover art finished. I cringed, because I wanted to be modest and avoid hyperbole, but Allen insisted I'd love it once I saw the cover.

The next time I visited Sebastopol to work on the layout, Michelle showed me a mockup of the cover, that used a black and white Norman Seeff photo that was taken for Rolling Stone magazine in January 1984, the one where Norman told us not to smile. They had colorized the Macintosh and added a bright red background. I liked the photo but disliked the red color, and was surprised to learn that my opinion didn't matter all that much. The O'Reilly team was adamant, telling me that I wasn't in a good position to judge, since it was supposed to appeal to young people, a group to which I no longer belonged. Eventually they wore me down and today I even sort of like it. At least I was able to get them to change "Revolutionaries" to "Revolution", which I thought seemed slightly more modest.

My biggest disappointment with the book has to do with the story links. We decided to keep the story links in the book, even though you can't click on them. To compensate, they were supposed to include the page number of the referenced story, but apparently that was too much for O'Reilly's layout system to cope with, given that page numbers changed frequently as edits were made. I hope we're able to improve this in subsequent printings.

Now that the book is complete, it's interesting to compare it to the website. Once I finally got a finished copy of the book in my hands, I was amazed at how much better it seemed than the website for continuous reading, in terms of ease and enjoyment, even though most of the content was crafted for the site instead of the book. Computers still have a long way to go before they match the ease of use of books. The website has some compensating strengths, though, and is better than the book for only reading stories about a particular character or topic.

But by far the main advantage of the website over the book is that it's a living document, capable of correcting itself and growing indefinitely. That might seem ironic, given the moribund state of the Folklore site since I finished writing in June 2004. But there's a (somewhat feeble) reason: I didn't want to write new stories while the book was in production, because I knew that I would want to squeeze them into the book, and I didn't want to delay it. Hopefully, this essay helped to shake off some of the rust, and I will start adding new stories soon, probably at the rate of around one per month.

You can buy **Revolution in the Valley** from Amazon by clicking here.

The Original Macintosh:

# REVOLUTION IN THE VALLEY
## Andy Hertzfeld

PDF convertion by Arianna Euryale 2011 (version 1.0)

Creative Commons

# Creative Commons License Deed

## Attribution-NonCommercial 1.0 Generic (CC BY-NC 1.0)

Disclaimer

The Commons Deed is not a license. It is simply a handy reference for understanding the Legal Code (the full license) — it is a human-readable expression of some of its key terms. Think of it as the user-friendly interface to the Legal Code beneath. This Deed itself has no legal value, and its contents do not appear in the actual license.

Creative Commons is not a law firm and does not provide legal services. Distributing of, displaying of, or linking to this Commons Deed does not create an attorney-client relationship.

This is a human-readable summary of the Legal Code (the full license).

Disclaimer

### You are free:

- **to Share** — to copy, distribute and transmit the work
- **to Remix** — to adapt the work

### Under the following conditions:

- **Attribution** — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- **Noncommercial — You may not use this work for commercial purposes.**

### With the understanding that:

- **Waiver** — Any of the above conditions can be waived if you get permission from the copyright holder.
- **Public Domain** — Where the work or any of its elements is in the public domain under applicable law, that status is in no way affected by the license.
- **Other Rights** — In no way are any of the following rights affected by the license:
  - Your fair dealing or fair use rights, or other applicable copyright exceptions and limitations;
  - The author's moral rights;
  - Rights other persons may have either in the work itself or in how the work is used, such as publicity or privacy rights.
- **Notice** — For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page.

A new version of this license is available. You should use it for new works, and you may want to relicense existing works under it. No works are *automatically* put under the new license, however.

What does "conditions can be waived" mean?

CC licenses anticipate that a licensor may want to waive compliance with a specific condition, such as attribution.

Learn more.

What does "Public Domain" mean?

A work is in the public domain when it is free for use by anyone for any purpose without restriction under copyright.

Learn more.

What does "Fair use" mean?

All jurisdictions allow some limited uses of copyrighted material without permission. CC licenses do not affect the rights of users under those copyright limitations and exceptions, such as fair use and fair dealing where applicable.

Learn more.

What are "Moral Rights"?

In addition to the right of licensors to request removal of their name from the work when used in a derivative or collective they don't like, copyright laws in most jurisdictions around the world (with the notable exception of the US except in very limited circumstances) grant creators "moral rights" which may provide some redress if a derivative work represents a "derogatory treatment" of the licensor's work.

Learn more.

What are "Publicity Rights"?

Publicity rights allow individuals to control how their voice, image or likeness is used for commercial purposes in public. If a CC-licensed work includes the voice or image of anyone other than the licensor, a user of the work may need to get permission from those individuals before using the work for commercial purposes.

Learn more.

Use this license for your own work.

This page is available in the following languages:
Armenian Belarusian Castellano Castellano (España) Català Dansk Deutsch Eesti English Esperanto français hrvatski Italiano Latviski Macedonian Magyar Nederlands Người Việt/Tiếng Việt Norsk polski Português Português (BR) română srpski (latinica) Suomeksi svenska Türkçe čeština Ελληνικά Русский српски العربية ไทย ქართული 中文 (香港) 日本語 華語 (台灣) 한국어