

To: Jonathan Software and Hardware

From: George Cossey

Date: July 25, 1985

Subject: Jonathan Software Compatibility with Macintosh.

---

## Macintosh compatibility and the Jonathan.

A primary goal of the Jonathan software is to be compatible with the current shipped Macintosh software, and stay as much compatible as possible with future versions of the Macintosh. Since the Macintosh ROMs were used as a base to start this project we have started with complete compatibility with that version of the ROM (which was somewhat incompatible with the current Macintosh).

Whenever possible, the Jonathan software will be made compatible with the Macintosh software. There may be cases where the software is incompatible if the application writer uses system calls in an illegal or extremely non-standard way.

There will always be differences in the software and the hardware that will cause some incompatibility problems.

*Some of the things that can cause incompatibility are the following:*

- a) Jonathan specific commands being added, mostly in the color area.
- b) Midi-Mac specific commands being added that do not apply to the Jonathan.
- c) Modules of different types and functions. Jonathan modules are more of a co-processor type, while Midi-Mac modules are more like the Apple II and IBM PC.
- d) Data structure differences. Jonathan records are a superset of the Macintosh records, color fields have been added for Jonathan. Jonathan supports both types of records with optimization done for the Jonathan size.
- e) Different CPUs result in exception frames being different. This means debuggers have to change. And in order to take advantage of the new 68020 commands, assemblers & compilers will have to change slightly.
- f) The interrupt structure is different. Jonathan interrupt levels are different from Macintosh because there are different devices that need to interrupt.
- g) The Front Desk Buss on the Jonathan makes the internal interface to all mouse, keyboard, time, and parameter memory routines different. The access at the hardware level is different, this will not normally affect application programs since they access these functions at a higher level.
- h) Hardware address and function differences will affect programs that use the hardware directly, such as some music programs and some copy protection methods.
- i) Handle addresses are changed from Macintosh. The Macintosh used the upper address bits as data holders for handles. Since the Jonathan uses all address bits as address bits, these bits are in a different place on the Jonathan. This area is currently being addressed.

j) The Jonathan is a fore-runner to more advanced operating systems that allow multiple applications to be running at the same time. To start making the transition to these operating systems, the Jonathan runs it's applications in the user mode of the 68020. This means that there are some instructions that an application program can not now use.

## **What does compatibility mean?**

Compatibility means that the Jonathan will run most Macintosh applications (see the compatability problems above). We will have to come up with a list of programs that will run when we ship. We should start contacting vendors that are using questionable programming methods now, the 68000 ROM group has a list of some of these, and request some changes for thier future versions.

Macintosh programs will run under the Jonathan PlayPen (Finder) and may not be able to boot themselves up (to be determined).

## **Traps.**

The traps being used on the current version of the Macintosh will be the same on the Jonathan. Additional traps being used on the Midi-Mac and on the new Macintosh ROMs may not be compatible due to the need of different functions on each system.

## **Sound, including voice.**

There are some, a small number, applications written by 3rd party people that use the sound buffer directly. We may not have to be concerned with them, only es far as trying to making sure that they do not clobber our O.S. if they are run. All other calls should be system level and we should be able to simulate the ones that make music. The ones that generate voice could be extremely hard to decode and simulate. Since we will have our own voice, and it will be much better, we should be OK.

## **PWM buffer.**

This is a speed control buffer used on the older single sided Sony disk drive. Since we are planning on using the double sided Sony drive, which has the speed control built on board, we will not have this function. So, we will not support an older type drive as our internal drive. Since modules can do anything, it is possible to place an IWM and a PWM buffer or equivalent on a module as an add-on board.

## **Single sided diskettes vs double sided diskettes.**

We will be capable of reading and writing in single or double sided format, both can be done on the double sided drive. By writing in a single sided format, our data disks could be used on the current Macintosh.

## **Serial communications.**

There has been a usable system level serial package available for the Macintosh from the start. We should have little trouble there, even if the SCC is located at a different memory location.

## **Copy protection on existing Mac disks.**

The more advanced copy protection schemes on the Macintosh are now directly talking to the IWM and the VIA chips. To fully support these is impossible with our hardware design. We will be compatible with the "Apple approved" method of copy protection, this is alternate address and data marks. Any other method that talks directly to the hardware will probably not work.

## **Finder.**

The Finder is really a DOS (Disk Operating System). In our environment we would need a device operating system that can handle all the modules that are being defined for the Jonathan, and all those that 3rd party people will develop. We need to have an idea of how each device could interface and be serviced by this device operating system. The Finder is currently used just to handle the disk drive and to launch programs. Our system will probably have to manage a lot more and may have a different "look".

## **New file system.**

The new file system will be used on Jonathan. This makes us compatible with current Macintosh diskettes as well as those that will come out under the new file system.

## **Hard disk, direct connect.**

Since we do not have an external disk connector (IWM style), we will not support the direct connect hard disk. There will be hard disk modules available for Jonathan.

Date: 07/29/85  
From: George Cossey  
Subject: What is Jonathon Software.

=====

This is an attempt to describe the current direction that the software for the Jonathan is taking. Suggestions, comments, and discussion are welcomed for all aspects of the software.

### Major features.

1. *Compatibility* with current Macintosh software to take advantage of existing software at introduction, while still expanding the software to a more advanced state. Designed to stay as compatible as possible with future products.
2. Easy *module* software add-on capability to the System software.
  - a) for Smart modules, acting as co-processors.
  - b) for Dumb modules, such as serial port and parallel port cards.
3. *Front Desk Bus* support, for all new devices on the FDB. Easy addition of new FDB devices as more are invented.
4. Advanced *Quickdraw*; including color and other features.
5. High quality color *images* as an integral part of the graphics.
6. New *file system* which better supports hard disks.
7. Larger *screen size*, 640 x 480.
8. *Color* modes without any decrease in screen resolution:
  - a) Black and White, 1 bit per pixel.
  - b) Four colors, 2 bits per pixel.
  - c) Sixteen colors, 4 bits per pixel.
  - d) 256 colors, 8 bits per pixel.
  - e) 65536 colors, 16 bits per pixel.

## Goals.

1. As *crash-proof* as possible. The 68020 allows for more recovery than the 68000. Also major changes in the ROM to make it more durable, such as recovering from more errors.
2. New scrapbook *format standards*, such as data (for spreadsheets and other number manipulators) and text.
3. Easy *batch mode* capability. Implementation of a standard way of using an exec file.
4. Incorporation of *cursor key* usage as an option to the mouse. The new keyboard has cursor keys.
5. Extended *graphics* capability with speed. More transfer modes. Animation graphics library.
6. Applications run in *User mode* for easy conversion to the next operating system which will use the MMU and be multi-tasking.
7. Add capability of *background tasks*, such as a print spooler, for external modules that require it.