

**Software Proposal**

**for**

**EPD NuBus Board Products**

**for use in the**

**Apple Macintosh II Computer**

by Pete Sucky

Monday, July 14, 1987

ELECTRONIC PHOTOGRAPHY DIVISION

---

## 1. Scope and Intent:

To propose some of the basic software that should be provided with EPD NuBUS boards (such as the framestore, the compander board and possibly a high speed modem or ESDN board) when sold for use with a Macintosh II computer. Through this proposal, hopefully, it should become clearer how much of an effort will be required, and where best to focus our resources so as to be most effective.

## 2. Macintosh Software Definition

The term "software" within this document refers to any code that provides control of the functions of our board level products above and beyond the ROM based or "firmware" code that actually resides on the board.

In the Macintosh environment there are many different types of software code that are accessed differently depending on their type. Some are more appropriate for controlling certain categories of products. Below is a list of common types of software in the Macintosh environment and a brief description of each.

### A. Chooser Level Drivers

These are files are placed into the system folder of the boot disk and are selected via an their icon which appears in the Chooser desk accessory. Some examples are the Laserwriter and Imagewriter printers, and several of the network server products such as AppleShare and NetModem.

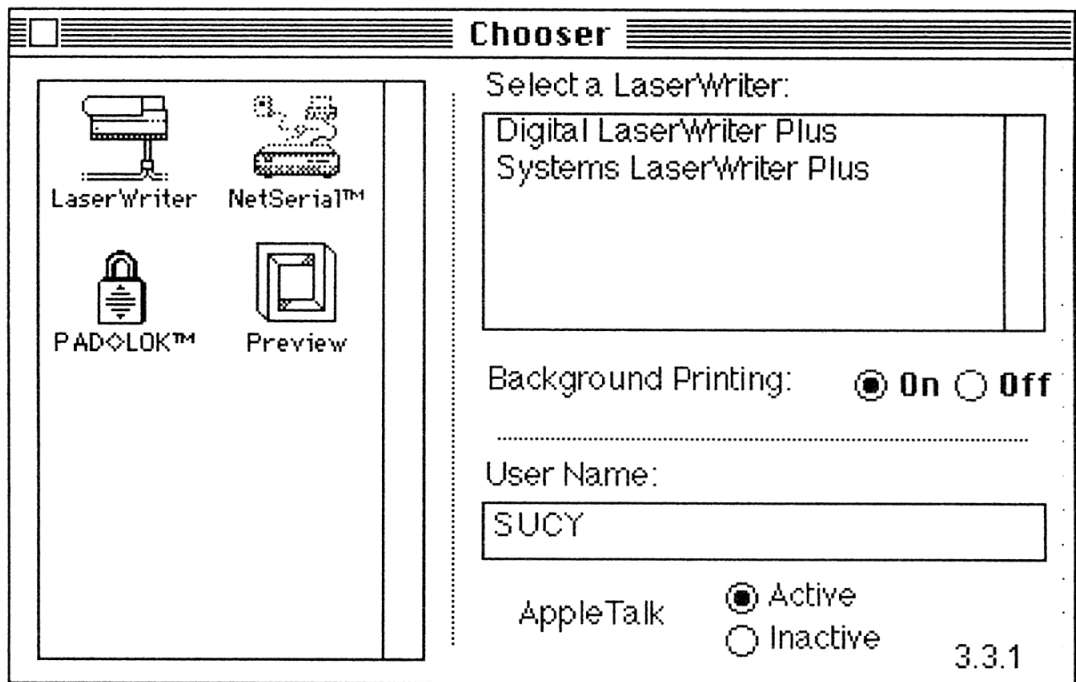


Fig. 1 Chooser desk accessory for selecting drivers.

## B. INITs (Initial Programs)

These are also located in the system folder and load during the boot up process, becoming memory resident applications and sometimes attaching to other resources to provide extended functionality. Examples of this are: Screensavers, Menuclocks, and TOPS. One public domain INIT called Quickfolder provides a button in all Save dialogs to create a new folder without returning to the finder.

## C. CDEV's (Control Device)

CDEV's are similar to INIT's (some INIT's are also CDEV's) in that they too reside in the system folder. CDEV's, however have options and controls that can be accessed via the control panel desk accessory as shown below in figure 1. Some examples are: General, Mouse, Keyboard, Monitors and Startup Device system files. Some commercial examples are Quickkeys™ and the RasterOps boards' CDEV.

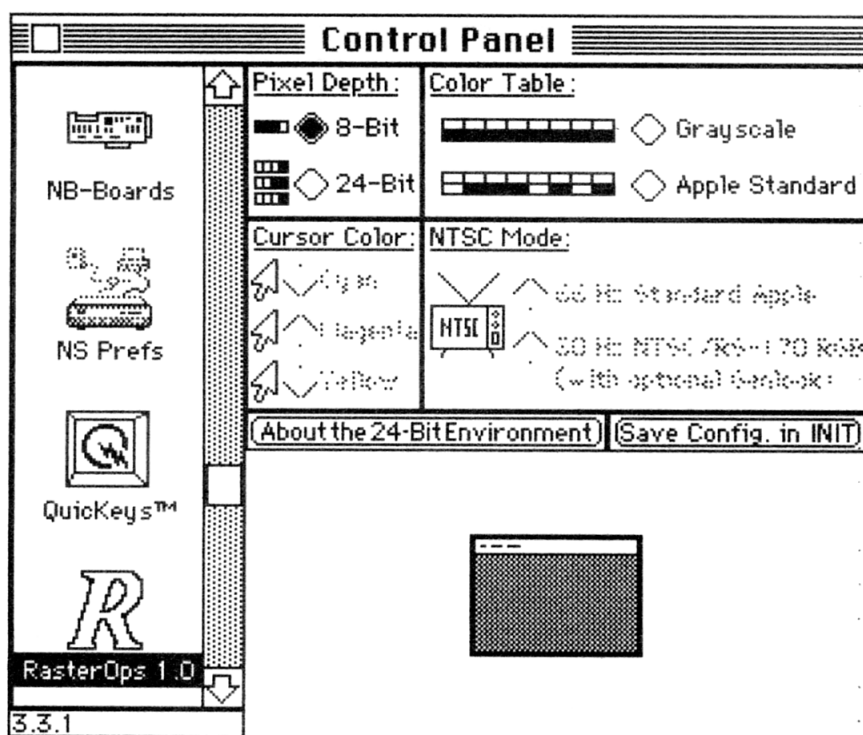


Fig. 2 Control Panel Desk Accessory with CDEVs on left.

#### D. Desk Accessories

These are small (1K to 200K) applications (a type of driver) that appear under the Apple menu. Since the Apple menu is common to all programs these programs are available in all applications. The control panel and the chooser are examples of desk accessories.

#### E. Hypercard and Hypercard Stacks

Since August 1987 Apple has been bundling a unique new program called Hypercard with all new Macintosh purchases. Apple has classified Hypercard as system software and as such it has been well supported with upgrades to date. It is however, difficult to classify Hypercard as just system software, it really has defined a new type of software category. In brief, it combines a word processor, graphics program, free-form database and a user programmable, object oriented, command language (Hypertalk) into a single program for creating, editing and accessing information. Basically it is a software construction kit.

The files created in Hypercard are referred to as stacks. The analogy here is to a stack of index cards. The Hypercard program is like an information recorder/player in that you can create your own information stacks or use it to play back and edit someone else's.

It has become a very popular way to distribute product information and is fast becoming the main search engine for CD ROM based storage on the Mac.

The addition of External Commands and External Functions (XCMD's and XFCN's) has extended the usefulness of Hypercard by providing access to Mac toolbox routines enabling just about any external control task to be implemented from within Hypercard by calling the command or function in Hypertalk.

#### F. XCMDs and XFCNs

Originally XCMD's and XFCN's were designed to extend the capabilities of Hypercard by allowing access to system calls and to external hardware. But it now appears that they are not just limited to Hypercard but can be used by other applications, if written that way. In effect they are software modules that can be shared by Hypercard stacks and other applications. Apple has distributed drivers for the Hitachi and Pioneer Laserdisc players in the form of XCMD/XFCN's for use with Hypercard. The preferred way of distributing XCMDs and XFCNs is by including them in a stack that shows examples of the commands and allows the user to cut and paste to create their own stack, customized to their way of working.

#### G. Applications

Applications are programs that are executable from the desktop by double clicking on the applications' icon or on a document identified as having been created by that application. Examples are: MacWrite, MacDraw, MS Word etc.

### 3. Proposal

As a minimum I feel it is necessary for us to provide XCMD's and XFCN's for controlling all of EPD's current serial interface products as well as the future board level products. These should be implemented in an informative and useful stack design that demonstrates the products' capabilities and provides the necessary information to allow the knowledgeable Hypercard end user to use the functionality of the supplied XCMD's in stacks of their own design.

Beyond making all of our products work in the Hypercard environment it appears that the XCMD's and XFCN's created for our products can be used as code modules for any future applications that we might develop.

As far as what these future applications might be, I think each product has to be looked at individually as to how it might be used in a typical workstation and then decide which of the software types described earlier best fits that function.

### 4. EPD Framestore

In addition to doing a Hypercard stack with all functions implemented, I believe the best approach for a useful product is in the form of a desk accessory. This will make the framestore available in all programs including the finder, thus not tying it to a specific application.

It is important to understand that as the framestore board is currently defined it will not serve as a Mac display screen. That is to say that desktop icons and such will not appear on the monitor connected to the EPD framestore. This is because the framestore will not work in non-interlaced mode and at a 35 KHz scan frequency.

The board will display a live video input, capture and display the captured image and then optionally transfer it to the standard Macintosh monitor

Below is a proposal for that desk accessory.

☐ **KODAK Framestore**

<b>Capture Mode:</b>	<input type="text" value="Live"/>	<div><b>Capture</b></div> <div><b>Move...</b></div>
<b>Display Mode:</b>	<input type="text" value="Frame"/>	
<b>Input Source:</b>	<input type="text" value="RGB"/>	
<b>Sync Source:</b>	<input type="text" value="Separate Sync"/>	
<b>Pixel Clock:</b>	<input type="text" value="768 x 484"/>	

Illustration fig. 5 below shows the mouse cursor over a pop-up menu, the mouse button is down.

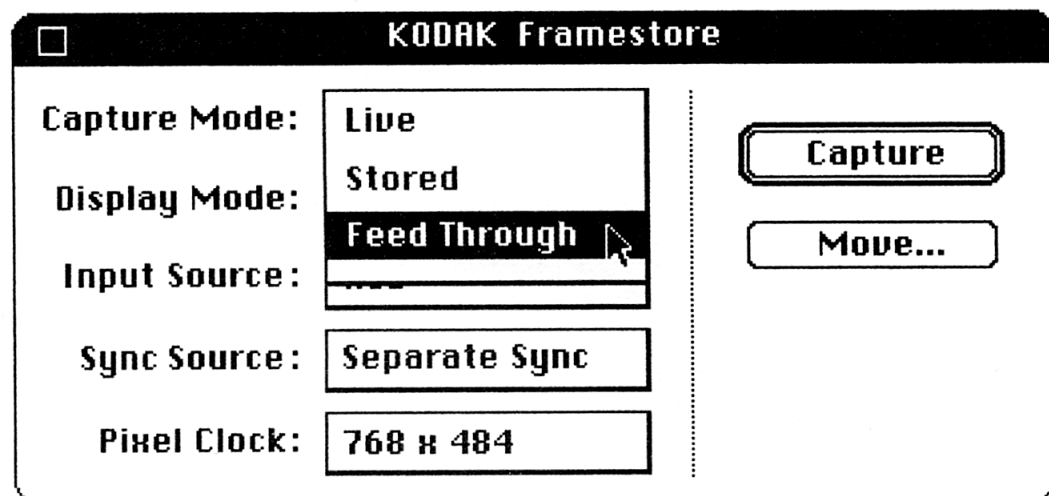


Fig. 5

#### Move Button Dialog box

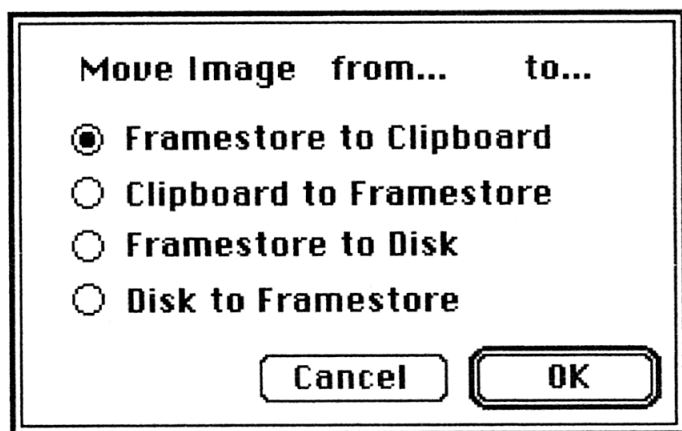


Fig. 6

The Move Button allows the user to designate where to send the image.

### **Capture Mode:**

**Live**  
**Stored**  
**Feed Through**

### **Available Menus and Options:**

### **Display Mode:**

**Frame**  
**Field 1**  
**Field 2**

The menus and options shown at left (figs. 4) are accessed by clicking and holding down the mouse button on the option to be changed, a pop-up menu will appear with the other available options listed.

Dragging the mouse to the new setting will hilite that option and make it show to indicate the current status after the mouse button is released.

### **Input Source :**

**RGB**  
**Y-C**  
**Composite**  
**RGB Color Seq.**

### **Sync Source :**

**Separate Sync**  
**Sync on Green**  
**Genlock**

### **Pixel Clock:**

**768 x 484**  
**640 x 480**