

Read Me First

**System Initialization for
ORCA/M V4.1**

**Copyright 1986
Byte Works, Inc.**

What to Do First

After making a major software purchase, you no doubt want to try it out right away. This booklet has several purposes, but its main one is to make your first experience with ORCA/M a pleasant one by helping you get started smoothly.

So why a separate booklet? Well, mostly to get your attention. A recent (and admirable) trend in consumer software has been in the direction of software that does not really need a user's manual, or at least doesn't need one until the advanced features are needed. Consequently, many people have gotten into the habit of jumping right in before looking at the directions. That's great for a program like Mousepaint, but doesn't work well for any product aimed at developing professional programs. ORCA is a very sophisticated program, and will take some time to learn to use. Like a fine tool or precision instrument, that time will be amply rewarded when the tool is put to use. What we want to do in this booklet is to help you unwrap the tool and plug it in. After that, your own prior experience will govern how you proceed. The other reason for a separate booklet is that we intend to continue updating ORCA, and these updates will probably effect how the system is initialized. When that happens, we only need to update this booklet, saving us money. As you will see when you begin to use the product, the savings are passed on in the form of far more program for the dollar than you will get from any other assembler, anywhere, on any computer.

Initial Set Up

What You Will Need

With your ORCA, you should have received the following:

- 1) This booklet.
- 2) A binder containing a sizable amount of documentation.
- 3) A 12 panel reference card.
- 4) Two double-sided disks.

To use ORCA, you must have an Apple // computer capable of running ProDOS. An Apple //c or Apple //e automatically satisfies this requirement. If you are using an Apple][+, it must have a 16K extension,

Read Me First

giving a total of 64K. ORCA will not work on an Apple II unless you have the autostart ROM installed.

ORCA is designed to work with eighty-column screens, and we strongly recommend using it that way. A forty-column driver is provided, but its main purpose is to give you something to use while you write a driver for an eighty-column card that we do not support. Currently, we support the Apple eighty-column card for the Apple IIe (this capability is built into the Apple IIc and Apple IIgs), the Videx Videoterm, and the Videx Ultraterm.

In addition to the items that came with ORCA, you will need four blank disks and your favorite disk copy program. Any copy program that will copy an entire disk will do – if you have no other, Apple supplies a simple disk copy program with both DOS and ProDOS, and either will do fine.

Backing Up the Disks

Let's start by being safe. Copy the disks that come with the system onto four blank disks, and label each of the disks appropriately. The ORCA disks are not copy protected, so, as we mentioned, any copy program will work. Once you have finished copying the program, put the original disks in a safe place.

This is a very important step – please do not skip it. The system initialization that we are about to do will permanently modify the /ORCA disk. If a mistake is made, or if your hardware ever changes, you will need to go back to the originals for a new copy of the disk.

Hardware Selection

ORCA/M is set up to work with a variety of hardware, including three eighty-column cards and any disk drive supported by ProDOS. This section tells you how to set up the initial bootable disk for your computer, which basically entails selecting an eighty-column card. Later sections will discuss other configuration options and how to get the most from large capacity disk drives.

The system initialization shown below will permanently modify your disks. Be sure and do this to a copy, not to the originals! You can use any copy program to make a backup copy of ORCA.

Apple //c or Apple //e with an Apple 80-Column Board

If you are using an Apple //c, an Apple IIgs, or an Apple //e with an installed eighty-column board, ORCA will boot and load as it is shipped. There are three drivers for other eighty-column boards that you can delete to make room for your own files. These drivers are located on the /LIBRARY disk in a subdirectory called DRIVERS. You can delete all of the files in that directory, as well as the directory itself. Be sure and do this to your working copy only – you may need to work on an older Apple some day, and the drivers could be useful then.

The Videx Ultraterm Board

If you are using a Videx Ultraterm board or an eighty-column board that is compatible with it, you will need to replace the ORCA.HOST file on your boot disk, /ORCA. Since you cannot boot ORCA yet, you must use another program, like Apple's filer (shipped with ProDOS) to copy the driver over. The steps necessary are:

1. Delete the file ORCA.HOST from the /ORCA disk.
2. Copy /LIBRARIES/DRIVERS/ORCA.VID2 to the /ORCA disk.
3. Rename /ORCA/ORCA.VID2 to be /ORCA/ORCA.HOST.

Once this is done, you can delete the contents of /LIBRARIES/DRIVERS, as well as the directory itself, to give yourself more work space. Your /ORCA disk is now bootable, and will work on any computer with a Videx Ultraterm board installed.

The Videx Videoterm Board

If you are using a Videx Videoterm board or an eighty-column board that is compatible with it, you will need to replace the ORCA.HOST file on your boot disk, /ORCA. Since you cannot boot ORCA yet, you must use another program, like Apple's filer (shipped with ProDOS) to copy the driver over. The steps necessary are:

1. Delete the file ORCA.HOST from the /ORCA disk.
2. Copy /LIBRARIES/DRIVERS/ORCA.VID1 to the /ORCA disk.
3. Rename /ORCA/ORCA.VID1 to be /ORCA/ORCA.HOST.

Read Me First

Once this is done, you can delete the contents of /LIBRARIES/DRIVERS, as well as the directory itself, to give yourself more work space. Your /ORCA disk is now bootable, and will work on any computer with a Videx Videoterm board installed.

The Forty-column Driver

If you have an eighty-column board that is not supported by ORCA, or if you do not have an eighty-column board at all, you must use the forty-column driver supplied with ORCA/M. This driver fakes an eighty-column board by showing the left or right forty-columns of the screen – whichever the cursor is on. This driver is provided primarily so that you will have a working version of ORCA/M to write your own eighty-column driver with.

To use the forty-column driver, you will need to replace the ORCA.HOST file on your boot disk, /ORCA. Since you cannot boot ORCA yet, you must use another program, like Apple's filer (shipped with ProDOS) to copy the driver over. The steps necessary are:

1. Delete the file ORCA.HOST from the /ORCA disk.
2. Copy /LIBRARIES/DRIVERS/ORCA.40 to the /ORCA disk.
3. Rename /ORCA/ORCA.40 to be /ORCA/ORCA.HOST.

Once this is done, you can delete the contents of /LIBRARIES/DRIVERS, as well as the directory itself, to give yourself more work space. Your /ORCA disk is now bootable, and will work on any Apple II computer with at least 64K and an autostart ROM.

Please note that the forty-column driver makes use of bank two of the language card area. It may not work with some hardware drivers.

Exploring the System

Before moving on, let's take a moment to explore the disks and get an idea how ORCA is set up. You will find that the CATALOG command works under ORCA. Give it a try right now.

What you see is a listing of the files on the system disk. Some of these are in subdirectories to better organize the disk. PRODOS, ORCA.SYSTEM, ORCA.HOST, and SYSCMND are used during the boot process, and are not actually needed when you are using ORCA. MONITOR is the program

Read Me First

that lets you type commands like CATALOG. It must remain available, since ORCA occasionally writes over it when a program is loaded that needs lots of room. EDITOR, SYSTABS and SYSEMAC are files used when you edit a program. EDITOR is the editor itself, and SYSEMAC is a file containing the editor macros currently defined by the system. SYSTABS is a file containing default tab settings for the editor. The LANGUAGES directory is where the assembler, called ASM6502, and the link editor are kept. The monitor, editor, assembler and link editor are automatically swapped in and out of memory as you ask ORCA to do things for you.

The next most commonly used disk is the /UTILITIES disk. If you have a two drive system, you should keep it in your second drive for now. ORCA is set up to allow commands which require external programs to be executed. In fact, you can even add your own commands. These external commands are called utilities, and live on the /UTILITIES disk. If you try to use one of the installed utilities when the /UTILITIES disk is not in a disk drive, ORCA will ask you to put the disk into a drive. (Hit the ESC key if you would rather abort the command.) Also on the utility disk is a HELP directory with a file describing the parameters and use of each of the commands in the monitor. If you can afford the disk space to keep these files in a drive while you learn the system, you will find it a much easier task. To use the help facility, you simply type HELP. That gives you a list of all of the commands available to you. Typing HELP followed by one of these commands will print the help file for that command on your CRT.

The last two disks contain the macro and subroutine libraries. The macro libraries consist of a large group of predefined macros that you can use when writing assembly language programs under ORCA. They are described in the Macro Reference Manual, located at the end of the documentation. Many of the macros make calls to library subroutines which are contained on the /LIBRARIES disk. These subroutines have already been assembled, so the system can bring them in automatically as part of the link edit step.

Getting the Most from Your Disks

One Drive Systems

ORCA can be used with a one drive system, and if you are writing programs shorter than about 1000 lines that do not use the subroutine

Read Me First

libraries, you will have no problems. Keep in mind that to use the utilities you will have to do some disk swapping. You will also have to do some disk swapping if you use the subroutine libraries or write longer programs. ORCA is unusual in the sense that if you do something that needs another disk, it will stop and ask for it. For example, you can split a program up into several sections, then put the source code on different disks. When the assembler finds an APPEND directive, it will ask for the disk it needs. By doing the disk swapping, you can assemble any program on a one drive system that could be assembled on a two (or more) drive system.

By the way, if you have an Apple //c or 128K Apple //e, you also have a half-size disk drive called /RAM. ProDOS automatically sets it up, and ORCA will happily use it. See the section on RAM disks to find out how to use it best.

Two Drive Systems

As you start to use ORCA, you should put the /ORCA disk in drive one and the /UTILITIES disk in drive two. When you are assembling a program that uses the subroutine libraries, replace the /UTILITIES disk with /LIBRARIES so that the linker can find the libraries right away.

After you get familiar with the system and start to write larger programs, you will want to start juggling things to keep from doing too many disk swaps. The best bet is to send the output to the /ORCA disk (or /RAM, if you have a RAM disk) and put the source code in drive two. If the source code extends over more than one disk, swapping won't be too much of a problem. Then, when the linker starts, you can replace the source code with the /LIBRARIES disk. It's more complicated to explain than to do, and will make more sense after you have learned enough about ORCA to be writing a program that big!

Small RAM Disks

If you have a RAM disk that is set up to work with ProDOS, there are a few things you should be aware of to make optimal use of it. First, the editor and some compilers will need some work space. They take this work space from a special prefix called the work prefix. Normally, the work prefix points to the directory where ORCA.SYSTEM was found (/ORCA as the system is shipped). To change it to /RAM, use the command

SET WORK /RAM

Especially in the case of the editor, which is interactive, the difference between using a RAM disk and using a floppy disk is very noticeable.

Another wise use of the RAM disk is to place any macro libraries there. ORCA only keeps four blocks of the macro library file in memory at any one time. The rest is handled very much like virtual memory on minicomputers. If the page faults are to the RAM disk instead of a floppy, things will be a lot faster. Even if you don't know what virtual memory is, just try assembling a program that has a macro file longer than five blocks. You will see the difference!

Hard Disks and Large Floppy Disks

Like a small child, ORCA will happily consume any resources available to it. If you are fortunate enough to be using a hard disk or a large capacity floppy, you will find that ORCA will have an entirely different personality. Naturally, you can keep all of the help files, utilities and libraries available all of the time. In addition, you will not need to do any disk swapping, even on large programs.

There is one peculiarity that we have noticed when using a Profile that we would like to pass on. Even if you have a RAM disk, you should leave the macro libraries for your program on the Profile. Although the RAM disk is certainly faster than the Profile, it turns out that putting the macros on the RAM disk doesn't save much time during the assembly. This is because if it hasn't been used for a while, the Profile parks its read/write head. Then, when it needs to do a read or write, it takes quite a while to move out to the track. If the assembler is constantly accessing the macro library, the Profile doesn't get a chance to rest, and the read/write head is positioned for business all of the time. The savings in track access time just about equal any savings you would get from putting the macros in a RAM disk.

To get ORCA onto your hard disk, start by creating the appropriate directory structure and copying the contents of the four disks to the Profile. Here at the Byte Works, we have found the following structure to be satisfactory (indentation indicates a subdirectory).

Read Me First

```
/PROFILE
  <stuff from the /ORCA root directory>
/SYSTEM
  <stuff from /ORCA/SYSTEM>
/LANGUAGES
  <stuff from /ORCA/LANGUAGES>
/LIBRARIES
  <stuff from /LIBRARIES>
/UTILITIES
  <stuff from /UTILITIES>
/HELP
  <stuff from /UTILITIES/HELP>
/MACROS
  <stuff from /MACROS>
```

After copying all of the files, delete the LOGIN file from the SYSTEM directory.

With this configuration, you can boot directly into ORCA when you boot your hard disk. If your hard disk is not bootable, you can boot into BASIC.SYSTEM and execute ORCA.SYSTEM on the hard disk.

If you do not want to boot directly into ORCA, you can place all of the above files into a directory on your hard disk. You should maintain the directory structure mentioned above. When you execute ORCA, it will look for certain files in certain directories, so the names mentioned above are critical.

Large RAM Disks

ORCA is too large to fit into a 64K machine without swapping parts of the system in and out of memory. Swapping takes time – the slower your disk drive is, the more time it takes. If you copy the system to a large RAM disk, you can significantly improve its performance.

The best way to copy the system to your RAM disk is by using the LOGIN file, an EXEC file that is executed automatically when ORCA starts up. This will only work well if you are using a hard disk or other large format disk, so that all of ORCA can be copied at once.

There are some parts of the system that you do not need to copy to the RAM disk. The files on the ROOT directory, and some of the files in the

Read Me First

SYSTEM directory, are used only at boot time. In addition, it takes a long time to copy the utilities and help files, so it is probably best to leave them where they are.

Once all of the files have been copied to the RAM disk, you must tell ORCA where to look for its files by using the SET command.

The following sample EXEC file assumes that you have copied ORCA to a hard disk called /PROFILE, as described in the last section. It further assumes that you have a RAM disk called /RAM that is large enough to hold all of the files. It creates the appropriate directories, copies the files, and then tells ORCA where to look for them.

```
CREATE /RAM/SYSTEM
CREATE /RAM/LANGUAGES
CREATE /RAM/LIBRARIES
COPY /PROFILE/SYSTEM/EDITOR /RAM/SYSTEM
COPY /PROFILE/SYSTEM/SYSTABS /RAM/SYSTEM
COPY /PROFILE/SYSTEM/SYSEMAC /RAM/SYSTEM
COPY /PROFILE/LANGUAGES/= /RAM/LANGUAGES
COPY /PROFILE/LIBRARIES/= /RAM/LIBRARIES
SET SYSTEM /RAM/SYSTEM
SET LANGUAGES /RAM/LANGUAGES
SET LIBRARIES /RAM/LIBRARIES
SET WORK /RAM
```

Some RAM disks, like those from Checkmate Technologies and Applied Engineering, must be initialized right after you boot. If you are booting directly into ORCA, you should include a call to the initialization program in the EXEC file. Both companies include an initialization program that can be executed from ORCA.

Once this EXEC file has been typed in, you should move it to the SYSTEM directory as the LOGIN file, so it will be executed automatically when you run ORCA.

Zero Page Usage

This section contains technical information needed by advanced users of ORCA. It is included here because it is subject to change. If you are new to ORCA, skip this section!

Read Me First

The ORCA.HOST shell is always in memory when you are using ORCA. It has hooks at \$FA to \$FF which must not be disturbed. The shell itself resides from \$800 to \$1FFF - the odd location was chosen to leave open the possibility of a version of ORCA for the Apple ///. Although the shell makes use of some zero page locations, it restores all of zero page to its original state. ProDOS does make use of a few zero page locations; ORCA/HOST does not restore these. See the ProDOS technical reference manual for details.

The subroutine libraries also make extensive use of zero page. The libraries are designed for compatibility with compilers, and work on the assumption that zero page locations \$80 to \$F9 are for their use. Any information stored in that area is subject to being overwritten. Currently, the libraries do not depend on the values of these locations between calls, so your program can also use the area for temporary storage. If you are using a compiler with ORCA, be sure and check the compiler's manual to make sure this is still true. Any locations from \$00 to \$7F used by the libraries are restored before control is returned to your program. Again, locations used by ProDOS are not restored.

Listing 1 shows the global common area used by the libraries. It is the definitive reference for exactly what locations are used. This same common area is copied into each of the subroutines libraries. You may also note some of the symbols showing up in your own programs - if a macro uses one of these locations, it first checks to see if the symbol is defined. If not, it defines the symbol itself, assigning the value from the global common area.

Read Me First

```
*****
*
*   COMMON - Common Global Equates
*
*   By Mike Westerfield
*   Copyright (C) September 1983, All rights reserved
*   By The Byte Works, Inc.
*
*****
*
COMMON START
;
;   Constants
;
FINF      GEQU  $20      bit maskS for flags in the sign byte
FZERO     GEQU  $40      of expanded floating point numbers
LUDP      GEQU  10      length of unpacked double precision
!          number
LPDP      GEQU  8        length of packed double precision
!          number
LUIFP     GEQU  7        length of unpacked floating point
!          number
LPFP      GEQU  4        length of packed floating point
!          number
!          bit masks for floating point errors:
FERR1     GEQU  1        invalid operation
FERR2     GEQU  2        division by zero
FERR3     GEQU  4        overflow
FERR4     GEQU  8        underflow
FERR5     GEQU  16       inexact
;
;   Zero page equates
;
SYSESTMIN GEQU  $94      evaluation stack min value
SYSESTMAX GEQU  $96      evaluation stack max value
SYSESTE   GEQU  $98      evaluation stack end
SYSCSTMIN EQU   $9A      call stack min value
SYSCSTMAX GEQU  $9C      call stack max value
SYSPSTMIN GEQU  $9E      program stack min value
SYSPSTMAX GEQU  $A0      program stack max value
SYSEOL    GEQU  $A2      end of line flag
SYSEOF    GEQU  $A3      end of file flag
WR0       GEQU  $C0      working registers
WR1       GEQU  $C1
WR2       GEQU  $C2
WR3       GEQU  $C3
WR4       GEQU  $C4
WR5       GEQU  $C5
WR6       GEQU  $C6
WR7       GEQU  $C7
MR0       GEQU  $C8      macro and parameter passing registers
MR1       GEQU  $C9
MR2       GEQU  $CA
MR3       GEQU  $CB
MR4       GEQU  $CC
MR5       GEQU  $CD
```

Read Me First

MR6	GEQU	\$CE	
MR7	GEQU	\$CF	
SYSS1	GEQU	\$D0	math registers
SYSE1	GEQU	\$D1	
SYM1	GEQU	\$D3	
SIGN	GEQU	\$E0	sign of result (used only be integer
!			math routines)
SYSS2	GEQU	\$E1	
SYSE2	GEQU	\$E2	
SYM2	GEQU	\$E4	
SYSFERR	GEQU	\$EB	
STR	GEQU	\$EC	location counter
END			

Listing 1

Editor Macros

ORCA gives you two kinds of macros. The assembler itself is a macro assembler, but the editor, like very expensive word processing systems, also has macros. These let you substitute a single keystroke for up to 128 predefined keystrokes. To use an editor macro, you hold down the closed Apple key and press an alphabetic key. (On the Apple][+, which does not have a closed apple key, you must use a game paddle button.)

All of this is explained in the reference manual, but the system comes with some predefined macros in the file SYSEMAC. Since these are subject to rapid change, they are described here. For each of the macros that are defined, the table below gives a brief description of what it is for. Trying them out on a blank file will no doubt help you to understand them.

letter	use
⌘A	draw a line of asterisks
⌘B	define a block comment
⌘C	define a normal comment
⌘D	<used by ⌘B>
⌘H	Byte Works copyright
⌘I	insert work space
⌘O	delete line to default buffer
⌘P	insert line from default buffer
⌘S	create subroutine header
⌘T	insert space in header

Read Me First

If You Have Problems

A program comment form is included here for any comments you may have about ORCA. We welcome any form of comment, be it constructive criticism, something you especially liked about ORCA, bug reports or documentation errors.

If you are reporting a suspected bug, please be very specific. We received one letter for DOS ORCA which said, "when you fix the bug in the directive, please let me know." We didn't know there was a bug, and had used the directive successfully ourselves. And of course the writer didn't even hint at what sort of problem he had had. We still don't even know if it was a bug or a user error! In any case, when you report a bug, you must supply us with enough information for us to see the bug. Until we see the problem, we really can't work on it. And, of course, be sure and tell us what hardware you have on your Apple. We aren't being nosy - we have seen several very peculiar bugs in software caused by a completely illogical conflict with a piece of hardware.

If you are having difficulty using ORCA, you can also call or write for support. Please understand that this support must be limited to answering specific questions. We cannot get involved in debugging your software.

You can contact us at:

The Byte Works, Inc.
4700 Irving Blvd N.W., Suite 207
Albuquerque, New Mexico 87114

Phone: (505) 898-8183