PROGRAMMERS AID

Rev 4.2

ADVANCED INSTRUCTIONS

Southern California
Research Group
Post Office Box 593
Moorpark, CA 93020

Telephone (805) 529-2082

quikLoader PROGRAMMERS AID
ADVANCED PROGRAMMING

THE C(hip) COMMAND

C specifies 2764, 27128, 27256, or 27512 EPROMs. If the C command is not used, the default size is 27256. There can be up to eight chip specifications in a source file.

The C command(s) can occur anywhere in the AID source file, but it is recommended that they always appear at the beginning for clarity.

If multiple chips are specified in the source file, the chips must be installed in adjacent sockets. Specify the chips in order from the lowest to highest socket number. In the example that follows, the 2764 could go in socket 6 and the 27128 in socket 7, or they could go in sockets 5 and 6, or any other adjacent pair.

```
C  2764
C  27128
K  {katalog name}
F  {disk file name}
```

Assume that this source file is named EXAMPLE. When the MAKE FILE is EXECed, the chip files will be saved on the disk as EXAMPLE.CØ and EXAMPLE.C1.

The first chip named always has a .CØ suffix, the second .C1, etc. If a 27512 is specified, then two files are generated. The files would be EXAMPLE.C1.BØ1 (bank Ø and 1) and EXAMPLE.C1.B23 (banks 2 and 3) if the second chip in the example were a 27512 instead of a 27128.

The AID will normally save an EPROM file on the same disk as the application files. With large chips or multiple chip applications, you might run out of disk space. You can avoid this problem by specifing a drive number with the CHIP command, e.g.:

```
C  27512, D2
K  My Program
F  B1
F  B2
F  B3
```

When the MAKE file is EXECed, the application files will be in drive 1, and the chip files (My Program.CØ.B01 and My Program.CØ.B23) will be stored at drive 2. (Remember to have an initialized disk in the drive.) This idea can be extended:

```
C  27256,D2
C  27256,D1
K  NEXT TRY
F  B1
F  B2
```

THE K(atalog) COMMAND FORMAT - K {name}

The K command begins applications and specifies the name that will appear on the quikLoader katalog. The katalog name can include upper & lower case, punctuation, and special characters. (Remember that an unmodified APPLE][or][+ cannot show lower case characters.) It can be up to 29 characters long.

THE F(ile) COMMAND FORMAT - F {name}

We will refer to the following example in discussing this command:

```
K  MYPROGRAM
F  FILE1
```

The F command causes the named disk file to be packed into the EPROM. It is moved from the quikLoader to RAM when MYPROGRAM is executed. FILE1 can be Binary, Integer, or APPLESOFT. Depending on the file type, the following happens:

APPLESOFT - The application is terminated. At execute MYPROGRAM time, DOS 3.3 is transferred to RAM, and FILE1 is transferred to RAM and executed.

Integer - The application is terminated. At execute MYPROGRAM time, DOS 3.3 and integer BASIC are transfered to RAM, and FILE1 is tranferred to RAM and executed.

Binary file - FILE1 is transferred to RAM, but the application is not automatically terminated. However, in this case, the application does terminate, since nothing follows FILE1. Since no special termination is specified in the source file, the quikLoader is turned off, and the motherboard is entered at the BSAVE address of FILE1. This is what you *want* to happen if FILE1 is a 6502 program whose first code is the start of the program, and which was BSAVED from its normal RAM location.

There can, and usually will be, more than one application in an AID source file. Refer to this example:

```
K  GRID          F  B1
F  G1            F  B2
K  BOAT
```

This source file will generate two applications on the quikLoader katalog, GRID and BOAT.

If G1 is a binary file, "K BOAT" causes GRID to be terminated with a jump to the BSAVE address of G1.

If B1 is APPLESOFT or Integer, the application is terminated with the quikLoader being turned off, and control going to the start of the APPLESOFT or Integer program. B2 is ignored.

Assuming B1 is a binary program, execution of BOAT causes transfer of B1, then B2. Program executation depends on which file type B2 is. If it is APPLESOFT or Integer, the execution is terminated, as above, If B2 is a binary program, motherboard execution begins at the BSAVE address of B2, the final binary file that was specified.

The above is an example of a very common disk application, where a BASIC program BRUNs or BLOADs one or more binary files. To place such an application on the quikLoader, *remove the BLOAD and/or BRUN commands from the BASIC program*, then specify the files in an AID source file as shown above.

Some binary files are not BSAVEed from the location at which they are normally RUN. A different RAM destination address can be specified:

```
K  BOAT
F  B1, A$1ØØØ
F  B2
F  B3, A$6ØØØ
```

In this example, B1, B2, and B3 are all binary files. At BOAT execution, B1 is transferred to $1ØØØ, B2 is transferred to its BSAVE address, B3 is transferred to $6ØØØ, then the quikLoader is turned off, and program execution begins at $6ØØØ.

THE G(omotherboard) COMMAND

Suppose, in the above examples, the start of execution address is not at the beginning of any of the disk files. Use the G command for this situation:

```
K  BOAT
F  B1, A$1ØØØ
F  B2
F  B3, A$6ØØØ
G  $1762
```

```
K  GRID
F  G1
F  B1
G  $1762
```

Flow goes to $1762 at the end of both of these applications.

Note that the file B1 is used in both BOAT and GRID. This does not waste space on the quikLoader EPROM, as the AID detects when a file is specified twice in a source file, and packs the file in only once.

THE P(ower-up) COMMAND

The P command marks the point where program execution begins at power-up if the chip is plugged in socket 6:

```
P  BOAT
F  B1
K  GRID
F  G1
```

This command does not have to be at beginnig of the file, it can replace any **K** command.

If a name does not appear after the **P** command, the following file(s) can only run as a power-up application, as the name will not appear on the katalog.

There can be only one **P** command in an AID source file. Extra **P** commands will be ignored.

If there is no **P** command in a source file, the standard power-up routine is installed. In the standard power-up, DOS 3.3 and Integer BASIC are transferred to RAM. This is the same thing that happens when no chip is installed in socket 6.

THE D(os) AND I(nteger) COMMANDS

If you are calling out a power-up routine, you may need to call out MOVE DOS (**D**) or MOVE INT (**I**). This will be done automatically with applications terminated by BASIC programs, but not binary files. Format is just the letters **D** or **I**.

RULES OF FLOW THE J(ump) AND L(abel) COMMANDS

FORMAT - J {name}
L {name}

K BOAT
F B1 (binary)
J GRID
K GRID
F G1

In this example, it is desired that BOAT not terminate (see STARTING AND TERMINATING, page 7) with B1, but flow instead into GRID. The "J GRID" command causes a JMP to GRID and inhibits autotermination. GRID execution causes transfer and execution of G1. BOAT execution causes transfer of B1 and G1, and execution of G1.

The {name} in J {name} must be the same as that in a **K** {name}, P {name}, or L {name} somewhere else in the source file. The L command is simply a means of installing a label that can be jumped to.

K BOAT
F B1
J DO G2
K GRID
F G1
L DO G2
F G2

GRID execution causes G1 and G2 transfer, and G2 execution. BOAT causes B1 and G2 transfer, and G2 execution.

THE H(ighRAM) COMMAND

QuikLoader transfers are usually accomplished with high RAM enabled for writing, bank 2. The H command enables transfer of data to bank 1.

```
K  BOAT
F  B1
H  $8B
F  B2
F  B3
H  $83
G  $DØØØ
```

Usually, before any transfer or entrance to the motherboard, a STA $CØ81 is performed. The H command usage in the above example results in the following high RAM control before quikLoader actions.

$CØ81 before B1 transfer.
$CØ8B " B2 "
$CØ8B " B3 "
$CØ83 before exit to $DØØØ.

The H command is usually not necessary. Use it only when it is desired to perform STA $CØXX control prior to quikLoader transfer.

THE S(ubroutine) and V(erbatim) COMMANDS

The AID performs all functions required to transfer files from the quikLoader to RAM and execute them. Sometimes, other actions may also be required. The V and S (do motherboard subroutine) commands can be of help with these other actions.

```
K  APA                    contains:
F  APA                    LDX #$87
V  APA.VBTM ------------  STX $74
S  $87E8                  LDX #$E1
G  $3DØ                   STX $73
```

In installing APA, it is found that locations $73 & $74 need to be set to $87E1 before APA execution. Also, it is desirable to execute an APA subroutine at $87E8 with the quikLoader off, then cold start DOS via entry at $3DØ.

Using an assembler or the mini-assembler, you enter the short program as shown above to any convenient memory location, then BSAVE it to disk in a file named APA.VBTM. The code sets $73 and $74 to the required value. The example above installs this code, verbatim, behind the 6502 routines that transfer APA to RAM. The "S $87E8" statement following "V APA.VBTM" causes motherboard subroutine $87E8 to be executed, with the quikLoader off, after the verbatim code. After subroutine execution, flow is returned to the quikLoader, where "G $3DØ" causes the quikLoader to turn off, and exit to the motherboard at $3DØ.

The verbatim file must be 6502 code with two very important restrictions. It must be completely relocatable, since you have no control over the address it will be executed from. Also, the accumulator must have the same value at the end of the code as it did at the beginning. In other words, if your verbatim code modifies the 6502 accumulator, it must eventually restore it to its entry value.

It may be impossible to live with the random location of the verbatim code. If so, just write the code to run at some RAM location, transfer it there via an F command, execute it via the S command, and continue quikLoader processing, as in this example:

```
K  APA                      S  $300
F  APA                      S  $87E8
F  APA.VBTM, A$3ØØ          G  $G3DØ
```

This example accomplishes the same thing as the previous one, but the code doesn't have to be relocatable, and the accumulator can be modified.

Just prior to execution of a subroutine via the S command, the contents of $45 - $49 of RAM are transferred to the 6502 ACC, X, Y, Status, and Stack pointer registers. After subroutine execution, the register values are stored at $45 - $49. A verbatim file can use this fact to transfer register values to and from the subroutine:

```
K  BOSTON WINDOW
D
V  WNDW.VBTM ──┐
S  $FDED       │
F  BOSTON WINDOW

               └─ LDX #$Ø
                  STX $36
                  LDX #$C3
                  STX $37
                  PHA
                  JSR $3EA
                  PLA
                  LDX #$AØ (space)
                  STX $45
```

In bringing up the binary application, BOSTON WINDOW, it is desired that DOS be initialized and 80 column mode be set. The verbatim file sets the COUT hook to $C3ØØ, notifies DOS (JSR $3EA), then sets location $45 to $AØ. Subroutine $FDED is executed with $AØ in the accumulator because "S" command loads register values from $45 - $49. The net result is that COUT ($FDED) outputs a space ($AØ)

NOTES: $3EA is a subroutine which can be executed with the quikLoader on, so it is called directly from the verbatim file (JSR $3EA).

The accumulator must be saved and restored in the verbatim file because $3EA modifies the accumulator.

Outputting the space seems to lock the 80-column mode in. It may not always be necessary, but it can't harm anything.

$FDED could not be called directly from the verbatim file because it is a monitor subroutine, and the monitor cannot be accessed while the quikLoader is on. Any subroutine that resides in, or accesses, memory external to the quikLoader in the $C1ØØ - $CFFF range must be called via the S command.

The decision to include the D command here is subjective. Including MOVEDOS clobbers any APPLESOFT program at application execute time, but it insures that the application will come up without fail. If MOVEDOS is not included, there will be times that Z-RESET must be performed prior to application execution.

N-RESET

The first application of every quikLoader source file is the N-RESET application for the resulting files:

```
K  BOAT  (this is N-RESET application)
F  B1
K  GRID
F  G1
```

If the first application begins with a K command, then that application can be executed via N-RESET or Q-RESET. If this chip is installed in socket 3, pressing CONTROL-3-RESET causes BOAT execution. Either BOAT or GRID can be executed via Q-RESET.

However, in this example:

```
F  A1
K  BOAT
F  B2
```

```
K  GRID
F  G1
```

A1 cannot be executed via Q-RESET. The only way to run it is via N-RESET. Again, BOAT and GRID can be seleted via Q-RESET.

STARTING AND TERMINATING

We will take this opportunity to list the various ways to start and terminate applications. The beginning of a source file is the N-RESET application. K and P commands begin applications.

The end of file and F(basic), G, and R commands cause quikLoader shutoff and entry to motherboard. The J command causes flow to deviate to another point of the quikLoader.

F(binary) opens the autotermination window so that following K or P commands cause exit to motherboard at the binary file address. In other words K and P files begin new applications and usually terminate the previous application unless it was already terminated by F(basic), G, R, or diverted by J. However, if there is no F command in an application, K and P will not terminate it.

LIST OF TERMINATING FUNCTIONS OF COMMANDS.

C	neutral
D	neutral
F(bin)	open window
F(basic)	terminate, close window
G	terminate, close window
H	neutral
I	neutral
J	divert flow, close window
K	Autoterminate if window open, close window
L	neutral
P	Autoterminate if window open, close window
R	Terminate, close window
S	neutral
V	neutral

EXAMPLES:

K	BOAT	N-RESET/Q-RESET ENTRY
F	B1(binary)	OPEN WINDOW
P		AUTOTERM, POWER-UP ENTRY, CLOSE WINDOW
I		MOVEINT, NEUTRAL
D		MOVEDOS, NEUTRAL
K	GRID	Q-RESET ENTRY
F	G1(binary)	OPEN WINDOW
S	$8ØØØ	NEUTRAL
K	SPLIT	AUTOTERMINATE, Q-RESET ENTRY
V	S2.VBTM	NEUTRAL
K	CAR	Q-RESET ENTRY
F	C1(binary)	OPEN WINDOW
F	C2(basic)	TERMINATE, CLOSE WINDOW

NOTES:

BOAT doesn't fall into POWER-UP because **P** autoterminates binary file.

POWER-UP flows into GRID because no files are specified in P.

SPLIT flows into CAR because no files are specified after SPLIT.

ADVANCED APPLICATIONS

CONDITIONAL BRANCHING

The AID J command actually generates a 3-byte 6502 JMP statement in the overhead file. This fact can be used in conditional branching during application execution. The testing is performed in a verbatim file which ends in a branch and is followed by an AID J command:

```
K   DOUBLE-TAKE 2.1                    TAX
F   DOUBLE-TAKE 2.1                    LDA #$FF
V   DT.VBTM ---------------------------PHA
J   SKIPVID                            PHA
F   ]I+VIDIO DRIVER,A$64ØØ             TXA
L   SKIPVID                            BIT $2C
G   $4ØØØ                             BMI +3
```

DOUBLE-TAKE is an application with a variety of 80-column video drivers. It is decided to build a quikLoader chip that will load the //e video driver if the quikLoader is installed in a //e, but will load the VIDEX video driver if it is installed in a][/][+. DOUBLE-TAKE analysis shows that the video drivers are $3ØØ byte files residing at $64ØØ - $66FF of DOUBLE-TAKE. It is possible to change the video driver by simply loading an alternate driver to $64ØØ before DOUBLE-TAKE execution. Execution of the above application loads the main DOUBLE-TAKE file which is configured with a //e video driver. The verbatim file tests whether the host computer is a //e or][/][+ (BIT $2C), and skips the following 3-byte JMP if it is a][/][+. The net result is that an alternate video driver is loaded if the computer is a][+.

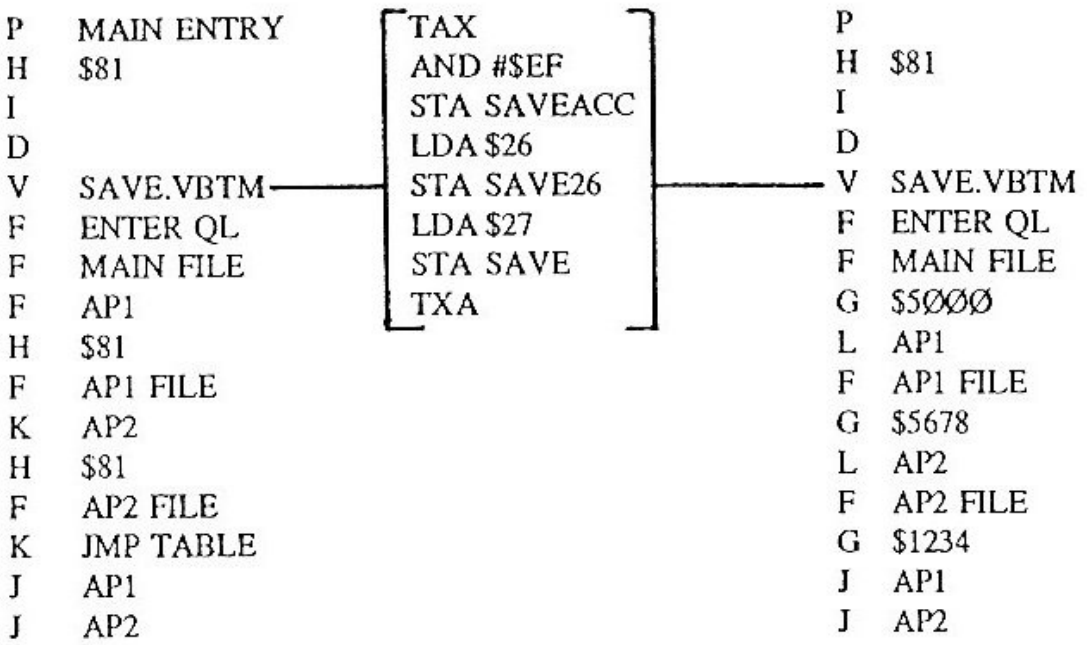
NOTES:

This location is always set up by QLOS at reset time. The MSB is set if the host computer is a][/][+ or compatible. $2C is reset if it is a //e.

It was found by experimentation that $FFFF needs to be pushed onto the stack before DOUBLE-TAKE execution. Sometimes, an application doesn't work when loaded from the quikLoader because some obscure action of the disk boot process did not occur. When this happens, you will have to trace it out and fix it, probably with a verbatim file.

JUMPING TO QUIKLOADER PROGRAMS FROM RAM

It is sometimes desirable to jump to a quikLoader application from a program in RAM. This can be fairly complex, because the RAM program must preset a number of parameters that are normally preset by QLOS. Some general examples are given here:



Both above examples accomplish the same purpose, except that no Katalog entries are generated on example B. The main program is entered at power-up (if plugged into socket 6), at n-reset, or at quikLoader re-entry. AP1 and AP2 can only be entered via the JMP table.

In example A, MAIN ENTRY is a quikLoader application which is capable of calling other quikLoader applications. The circumstances under which the quikLoader is re-entered are determined by the programmer. We assume here that, sometime after MAIN ENTRY has been executed, the need exists to execute AP1, AP2, or MAIN ENTRY.

SAVE.VBTM is a verbatim file that saves critical values for re-entry to the quikLoader in a safe location. This location can be any that will not be destroyed by the applications.

NOTES:

All applications begin with "H $81". This is necessary on a re-enterable program in case an applicatiion has modified location $112 since the last reset. Substitute other H values if your application requires it.

The SAVE.VBTM file saves ACC, $26, and $27. It is not likely, but possible, that $2C (]|-//e flag) and $2D (quikLoader slot map) will also need to be saved. This depends on the program.

The JMP table does not have to begin with a K statement. K JMP TABLE is only there to terminate AP2 in case AP2 FILE is a binary file. AP2 could also have been terminated with a G command.

The JMP table gives entry points to AP1 and AP2. If the JMP table is at the end of the source file, the JMPs to AP1 and AP2 will be located as follows:

27128, 256, 512, and 64 with no GETSLOT overhead	2764 with GETSLOT overhead
$FFD4 JMP AP1	$FF4D JMP AP1
$FFD7 JMP AP2	$FF5Ø JMP AP2

The JMPs are located at $FFD7 (or $FF5Ø) and backwards in steps of three bytes. The location of each jump is listed during pass 3 of AID assembly.

No JMP to MAIN ENTRY is necessary since MAIN ENTRY is the N-RESET routine. There is always a JMP to N-RESET/POWER-UP stored at $FFEF of the quikLoader. CLC, JMP $FFEF enters the POWER-UP routine. SEC, JMP $FFEF enters the N-RESET routine. In Example A, N-RESET and POWER-UP are the same, so that the carry flag is not important.

Assume now that MAIN ENTRY operation determines it is time to enter AP1, AP2, or MAIN ENTRY. A jump is made to the ENTER QL program. The following is a sample version of ENTER QL. It can be stored at any conveniant RAM location that doesn't interfere with the applications.

```
1   LDA  $CØ81
2   LDA  $CØ81
3   LDX  #$EØ
4   TXS
5   STA  $CØØ7
6   STA  $CØØA
7   LDA  SAVE26
8   STA  $26
9   LDX  SAVE27
10  STX  $27
11  LDA  SAVEACC
12  STA  $CØ8F,X  quikLoader ON
13  JMP  (adr) go to application
```

NOTES ON ENTER QL:

Lines	1 - 4	Not always necessary, but usually a good idea.
"	5 - 6	Possibly necessary if APPLE //e
"	7 -13	Always necessary
Line	13	Your program must set up the indirect address, probably to $FFEF, $FFD7, or $FFD4.

ANOTHER EXAMPLE :

```
K  AP1
F  AP1 FILE
K  MAIN ENTRY
F  ENTER Q
F  MAIN FILE
P  AP2
F  AP2 FILE
```

This example is not meant to reside in socket 6, even though it specifies a power-up applicaion. Specifying a power-up application is just a tricky way of generating two selectable applications at $FFEF. The ENTER QL routine can jump to AP1 via SEC, JMP $FFEF. It can jump to AP2 via CLC, JMP $FFEF.

GETSLOT AND FALL OVERHEAD

The AID normally packs GETSLOT and FALL overhead into quikLoader EPROMs. GETSLOT overhead takes up $45 bytes in the CØ chip. FALL overhead takes up $11 bytes in all but the highest chip. There is no FALL overhead in one-chip applications

When GETSLOT is included, that chip can be used in socket Ø of extra quikLoaders in multiple quickLoader applications. QLOS needs to reside only in socket Ø of the highest priority quikLoader (lowest slot number).

FALL overhead enables N-RESET to fall through to a lower priority quikLoader when a chip has no N-RESET routine. This is advantageous because there is only one N-RESET routine per chip in multiple chip applications. (The N-RESET of the highest numbered chip socket of the application.).

GETSLOT and FALL overhead are of no benefit in computers with only one quikLoader. If AID assembly has a small overflow that might be cleared up with GETSLOT and/or FALL omitted, it attempts re-assembly without it/them. Predicting the effects of GETSLOT/FALL removal is inexact, so AID might actually re-assemble the source file twice before succeding or giving up.

MODIFYING THE PROGRAMMERS AID

The AID is an unprotected APPLESOFT program. Users may wish to improve or change it, and are encouraged to do so, and report improvements to us.

There are some program defaults which you may find need to change. For instance, the maximum number of lines in a MAKE file is 200. You can change this with the DIM statement at line 32012.

Overhead file is made up of Katalog entries, EXECQL parameter lists, and 6502 routines (primary routines). Work spaces of 1024 bytes are set aside for each of these areas. Overflow is possible, especially with the primary routine space which must hold all verbatim files. You may modify the following lines when overflows occur:

Primary Routine Overflow	31102
Parameter list overflow	31103
Katalog overflow	31104

Line 31120 sets the standard source and destination drive suffixes for the MAKE file. The source drive is S6,D1 (DS$="S6,D1"). The chip file destination drive is S6,D1 (DD$="S6,D1"), unless changed by the C command.

BIG HINTS!:

Use a program optimizer to remove remarks from AID to make it run faster and take less space.

GETSLOT and FALL overheads can get in the way when you want files to be packed in a orderly fashion. To force omission of GETSLOT and FALL, add the lines:

```
31Ø95 FA=Ø:GE=Ø:F9=FA
and 32013 GOTO 32Ø3Ø
```

ASSORTED INFORMATION

The G, S, H, and F {name},(addrs) commands can have a decimal or hexadecimal argument. "G $8ØØ" is the same as "G 32768". Just remember to preceed hex arguments with the dollar sign ($).

Maximum length of F command files is $88ØØ (34816). (subject to increase).

After a D command, it's possible to cold start DOS and enter APPLESOFT via a G $3DØ command. It's also possible to enter DOS and execute a DOS/APPLESOFT command such as CATALOG or RUN.

```
K  BOAT
D
F  B1 (binary)
V  BOAT.VBTM  --- (see below)
G  $9FB3
```

For reasons known only to him, the programmer has taken a perfectly good APPLESOFT program, and converted it to a binary file. Now he wants to get it up and running as an APPLESOFT file. BOAT.VBTM looks like this:

```
LDX #$B6          set FP pointers
STX $69           set end of program to $BB6
STX $AF
LDX #$OB
STX $6A
STX $B0
LDX #$D2          "R"
STX $2ØØ
LDX #$D5          "U"
STX $2Ø1
LDX #$CE          "N"
STX $2Ø2          "RUN" to $2ØØ - $2Ø2
LDX #$8D
STX $AA5C         CR to $AA5C
LDX #3
STX $AA5A     length of "RUN" to $AA5A
STX $AA52     DOS video intercept state to 3
```

The above example enters DOS and executes á "RUN" command. Since the program was transferred and FP points set, the FP program will RUN.

TO PLACE AID ON A QUIKLOADER EPROM

1) Use an APPLESOFT optimizer to remove all remarks and shorten all variable names.
2) Delete line 31045
3) Run AID with the following source file (or add this to another source file):

```
K quikLoader AID
F CAT
F AID
```

RESERVED FILE NAMES

The following disk file names are reserved while running AID:

AID	GETSLOT	FALL
CAT	FPMOVER	

and, if your source file is named NAME:

NAME.Cn	NAME.Cn.B23	NAME.TEMP
NAME.Cn.BØ1	NAME.OVHD	MAKE.NAME

HINTS AND TECHNIQUES

As AID is an APPLESOFT program, press CTRL-S to pause during execution, and press CTRL-C or RESET to terminate execution.

Enter "PR#1" before running AID for a record of the application. Make sure that the printer is connected and powered on. Or enter "MON I,O,C" and "PR#1" before running AID for a REALLY complete record of the application.

Enter "PR#1" before you EXEC the MAKE {name} file for a record of the MAKE process.

To analyze OVHD file:

1) Get BLOAD address from MAKE file.
2) BLOAD NAME.OVHD to that address.
3) CALL -151
4) GET N-RESET FROM 8FFØ.8FF1 (subtract S8ØØØ, e.g. SFF23 = S8F23)
5) View KAT record via BLOAD.NRESET-1 CR. Everything up to S86 is KATALOG; Everything after S86 is EXECQL parameter list
6) NRESETL CR lists primary routines.

Primary routines normally run up to SFFDA, or to SFF53 in a 2764 with GETSLOT included. In all OVHD files, SFFDA - SFFFF is standard except NRESET and KAT pointers at S8FFØ/1 and S8FF8/9. If standard power-up routine is included, its EXECQL parmlist will be a S8FD1 - S8FD5, and its primary routine will be at S8FD6 S8FD9.

SUMMARY OF COMMANDS

COMMAND	FORMAT	DESCRIPTION
C	C [chip type]	Sets up file to use different size chips. Default size is 27256. Use 2764 through 27512.
D	D	Move DOS 3.3 to RAM.
F	F [file name]	Name of file on disk that the AID looks for.
G	G S[addrs]	GO TO MOTHERBOARD. Turns off quikLoader.
H	H S[addrs]	HIGH RAM CONTROL. Causes STA at $CØ[addrs]
I	I	Move Integer BASIC to high RAM. Not necessary to run normal integer programs.
J	J [name]	Jump to file named [name]. This can be a K, P, or L file. Does not turn off quikLoader.
K	K [name]	The file name as you want it to appear on he quikLoader Katalog.
L	L [name]	Sets a Label for the J command.
P	P [name]	Tells the quikLoader that this is the power-up application. NOTE - If [name] is left blank, the following file will run only at power-up, as it will not show on the Katalog.
R	R	Turns off quikLoader and performs motheroard Reset
S	S S[addrs]	Turns off quikLoader, does a Subroutine on the motherboard at [addrs], and goes back to the quikLoader. $45 - $49 = Acc, X, Y, status, and stack pointer.
V	V [name]	Verbatim. Takes the file [name] and puts it verbatim (that is, exactly as is) into the quikLoader primary routine. The file must be 6502 machine code. The accumulator value must be retained.

Deckard was here.