

NOTICE ON AMPL/M RELEASE 3.2

*** PLEASE READ FIRST! ***

1. Description of Current Release

This release has many new features, and as far as we know, all bugs corrected. Below is a complete list of the new features, as well as the corrected bugs. New features which need explanation are covered in detail. NOTE: Any program generated from this version of AMPL/M CANNOT be read into computer memory by a previous version of AMPL (before version 3.2). If you attempt to do so, you will get a "NOT AN AMPL/M FILE" error message (when this problem occurs with version 3.2 in the future, the error message will be "CANNOT SUPPORT NEW AMPL/M STATEMENTS"). This is to protect you from crashes due to illegal statements from later versions being loaded into earlier versions. Any older file CAN be read in by this version, however.

1. The long-awaited Multi-Image Utilities are complete. You can now "upload" from a Star-3 Memory Dump or directly from a Star-3 program tape into the Superstar memory. If you use a Star-3 Memory Dump, your cues will include all repeat loops, etc. If you use a program tape, all wait times will be automatically calculated between cues. All waits will be expressed as "WAIT X = nn.nn" statements, so editing is easier.

The directions for using these two utilities (BUILD FILE FROM TAPE and LOAD STAR-3 MEMORY FORMAT) are shown on the monitor when either is selected from the Multi-Image Utilities Menu.

2. The EXECUTE cue utility is now working (also from the Multi-Image Utilities Menu), as well as the EXECUTE cue decode function in RUN mode. Any EXECUTE cue being decoded during RUN will act exactly like a remote cue or a keyboard cue. The system clock will blink briefly in reverse video on each EXECUTE cue.

This feature makes the Superstar compatible with Star-3 Memory dumps which are started by EXECUTE cues.

3. A new statement "CLOCK PICKUP: T = H:MM:SS.FF" has been added.

To enter this statement, type <CP>, followed by a <RETURN> if you want the default time of 0:00:05.00. Otherwise, type in the desired time. The specified time will be the time at which the clock pickup will occur. The default value is useful if you are starting a tape segment whose clock value

begins at 0:00:00.00.

This statement can be used on any time line, and will place a message on the message line "START NEXT TAPE SEGMENT". While the time line in which the statement is used will "hang" until the clock pickup takes place, the other three time lines can be progressing as usual. When the clock track is decoded, if the clock time on the tape is past the pickup point, the message "WARNING: TAPE PAST PICKUP POINT" will be displayed. Otherwise, a countdown to pickup will be displayed, and when the pickup occurs, the system showtime (time displayed in the mode window) will be instantly changed to the pickup value, and all time lines will proceed. Note that while the clock value will make a sudden value change, this WILL NOT affect the relative timing of the other three time lines. Animation effects will not display any "glitches".

This statement is the "state of the art" for both speaker support with canned sections, as well as continuous showings, where the Superstar controls the tape recorder. There are several excellent recorders on the market with "rewind to zero" and "start/stop" by remote control. By using the Clock Pickup feature at the beginning of a loop, and by using auxiliaries to rewind and restart the tape recorder, you can have a continuous show running automatically. Also, if you are clever with your programming, you can have slides projecting on the screen during the tape rewind section.

4. A new editor command <BA> has been added. This is the "Block Adjust" command, which allows you to move all WAIT UNTIL T values in a block forward or backward in time. Simply mark a block, and type <BA>, then a <+> or <->, depending on which way you want your WAIT UNTIL T's to be adjusted, and then a H:MM:SS.FF value. When you hit <RETURN>, all unlocked WAIT UNTIL T's in the block will be adjusted. If you make an adjustment such that a T value exceeds 2:39:59.99, or goes less than zero, an error message "ADJUSTED TIME OUT OF RANGE" will be given, and the offending cue will be displayed as the current cue. Note that if this error occurs, the adjustment will not be made on any cue after the error in the specified block.
5. A new UNTIL statement has been added, along with powerful new versions of the original UNTIL. The new statement is UNTIL X, and uses X type variables rather than C type variables. The tests available on the two UNTIL statements are shown below:

UNTIL Ci = nnn	UNTIL Xi = n.nn
UNTIL Ci = Cj	UNTIL Xi = Xj
UNTIL Ci > nnn	UNTIL Xi > n.nn
UNTIL Ci < nnn	UNTIL Xi < n.nn
UNTIL Ci # nnn	UNTIL Xi # n.nn
UNTIL Ci > Cj	UNTIL Xi > Xj
UNTIL Ci < Cj	UNTIL Xi < Xj
UNTIL Ci # Cj	UNTIL Xi # Xj

Where i, j are any digit 0-9
 nnn is any number from 0-999
 n.nn is any number from 0-9.99
 and > is "greater than"
 < is "less than"
 # is "not equal to"

To enter these statements, simply type either <UC> or <UX>, followed by the operator (>, <, #, =) and a value or variable name. Then, press <RETURN> or <SPACE>. Entry is similar to the LET statements.

You can still use the original default for the "UNTIL CN = 0" as well as "UNTIL XN = 0" statements. Type <UCN><RETURN> or <UXN><RETURN> for these special cases.

With the above added capability, you no longer have to use your calculator to figure out your loop end tests. For example, to do a speed-up loop, the following is possible:

```
LET X1 = .25
REPEAT
  UNTIL X1 < 0.05
  CUT 1
  WAIT X1
  LET X1 = X1 - .01
END: REPEAT
```

The loop above will terminate when the speed of animation reaches 20/sec. If you want a LINEAR speed up (the above is exponential), use two time lines, with the following two loops running:

timeline #1	timeline #2
-----	-----
LET X1 = .25	REPEAT
REPEAT	UNTIL X1 < .05
UNTIL X1 < .05	CUT 1
WAIT 0.25	WAIT X1
LET X1 = X1 - 0.01	END: REPEAT
END: REPEAT	

Timeline #1 changes the wait time in a linear fashion (not faster as the loop speeds up), and timeline #2 uses the linearly changing value to control its animate speed. Both loops will terminate at the same point, in this example.

Another use of these statements is to create IF-THEN-ELSE constructions. To do this, see the example below:

already a single character entered on the command line in the editor. Usually, this letter is F, for some reason. To fix this, the editor will accept any of the four control codes specified above after a character has already been entered on the command line. The character will be thrown out, and the control command executed, with no error beep.

10. Other beep problems fixed: The first relates to using <CTRL-F> or <CTRL-R> several times in a row. Previously, if you entered another page forward or reverse command while the first was still being executed, you would get the error beep. No longer!

The second relates to use of <TF> and <TR> commands. No longer do they beep when you tab to the beginning or end of the program. Ah! Sweet silence!

11. All "run time" error messages now are properly reported, and will stop your program from running when they occur. The current line pointer will point to the the NEXT cue in the program, after the one which caused the error. If you press <CUE> to continue, you will get a "RUN TIME ERROR" because the computer cannot continue until you fix the problem. Each of the run time errors is described below:

MORE THAN 4 REPEAT LEVELS

Occurs if you try to nest more than 4 levels of repeat loops on a single time line.

UNINITIALIZED VARIABLE

Occurs if you use a X or C variable without first defining its value with a LET statement.

INDEFINITE REPEAT WITH NO UNTIL

Occurs if you have an indefinite repeat loop (the REPEAT statement has no value) and have no way for the computer to exit the loop. An indefinite loop must have at least one UNTIL statement, to allow exit.

C VALUE EXCEEDS 9999

Occurs if you allow a loop you are syncrolinking to run more than 9999 times.

REPEAT WITHOUT END

Occurs if you forget the end of a loop, and the computer tries to find it to do an UNTIL exit during the first run through the loop. If the computer executes to the end of the program the first time through the loop, you will not get this error, but rather the END OF PROGRAM message.

WAIT XN VALUE = 0

Occurs if you try to use a value of 0.00 in a WAIT XN statement. Zero waits are not allowed.

ARITHMETIC OVERFLOW

Occurs if you add to a C or X variable in a LET statement, and cause the value to exceed 9999 or 99.99.

ARITHMETIC UNDERFLOW

Occurs if you subtract from a C or X variable in a LET statement, and cause the value to go negative.

SHIP BUFFER OVERFLOW

Occurs if your program generates cues faster than the specified SPEED for more than 16 cues in a row.

TASK NAME NOT FOUND

Occurs when a START TASK: statement is executed, and the specified task name has not been defined.

NO TIME LINES AVAILABLE

Occurs when a START TASK: statement is executed, and four time lines are already in use (the main time line plus three other tasks running).

12. Tabs are now allowed inside loops. This allows you to use tabs even if your entire program is a loop (when using continuous shows with CLOCK PICKUP).
13. The print utility has been improved in two ways. First, you can print only a block of cues, rather than the entire file. Simply mark the desired block in the editor, and answer the "PRINT BLOCK ONLY" prompt with a <Y>.

The second improvement relates to compatibility with various printer interface cards and printers. Up to now, AMPL only supported the Apple and Epson cards. Now, AMPL will support just about any card. YOU WILL HAVE TO ENTER THE PROPER CONTROL STRING to make this happen. The default control string is the one which works on the Apple and Epson cards, so simply press <RETURN> to accept the default when asked for the control string. If you have some other card, you will have to read the manual to see what control characters are needed for your specific printer and card. If you are lucky, it will be the same string! Once you enter the string, you can use the printer over and over, without retyping it. You WILL have to retype the string each time you boot the system, however.

To be AMPL compatible, the card/printer combination should not echo characters to the Apple CRT screen, should operate with 80 column lines, and should generate a line feed

automatically (either the printer OR the card, not both!). The control string you would have to enter for the Apple or Epson cards is <CTRL-I><BON>. This turns off screen echo, and sets up for 80 column operation. This string would be entered as <^IBON> on your keyboard. The "upcaret" is a shifted N on Apple II Plus computers, and is used to indicate that the next character is to be treated as a control character. DO NOT ENTER THE CONTROL CHARACTERS THEMSELVES. Of course, you don't have to type this line if you are using either of the two standard cards.

Another example of a control string that you might need is <CTRL-I><K>. On the standard Apple and Epson cards, this turns off the automatic line feed at the end of each line. This is necessary only if your printer generates its own line feeds, and you can't shut them off with switches on or inside the printer. To get this feature, the entire string <^IBON^IK> would be required.

One other important item: many printers have "escape" sequences for controlling special functions, such as enhanced print mode, or compressed mode, or double strike. These modes are signaled by an escape character followed by some other character. The Epson MX-80 uses ESC-E for enhanced mode, for example. To get this mode, add the ESC-E to the control string. To do this, you must use the "upcaret" control convention described above! If you try to type an <ESC> into the control string, AMPL/M will "escape" the print utility, and return to the main menu. A <CTRL-[> is an escape. The control string must thus have <^[> in it. To get a "[" character from an Apple II plus, you must use the <CTRL-S> special character shift function (shown in the Editor Insert Menu). Type <CTRL-S> followed by a "<" character to get the "[" character. The complete control string would be <^IBON^[E>.

14. Tray home position now defaults to one, not zero. You can change it with a single keystroke to either <1> or <0>.
15. The ESCAPE statement has been added to AMPL. This statement is a debugging tool. When the computer executes this statement, an escape will be executed, just as if you had pressed the <ESC> key yourself. This can be used in tasks or on the main time line, and allows you to set up a stop for the computer at some point in your program. Enter the statement by typing <ES>.
16. There is a new default for the WAIT UNTIL T statement. If you are in RUN mode, and wish to enter the current clock time into a WAIT UNTIL T statement, simply type <WT?> (in the insert or edit mode, of course). The current time will automatically be entered.

This feature is designed to simplify editing. If you want to work on a sequence in the middle of your show, go to the end of the sequence, autosync, and insert a WAIT UNTIL T statement using this feature. Now, if you change the total time in the previous sequence, the next sequence will not be

affected.

17. A complete new status display has been added. This new display shows many new pieces of information, which will make it easier to debug your programs. The new items are:

VARIABLE VALUES:

All 10 C and X variables are displayed, along with their current values. The dash (-) is used to indicate an undefined value (no LET statement has set the value so far in the program). This will allow you to see what is happening with your variables, and also to figure out which ones you are using in your program (to do this, autosync to the end of your program, and then look at the status display). All values are undefined when RUN mode is off.

CURRENT HOME POSITION:

The current home tray position is displayed at the bottom of the screen. You can enter a <1> or <0> from the keyboard to set the home position.

MAIN CUE:

In RUN mode, indicates the current cue being executed on the main time line. If a WAIT is being executed, the cue number will be for the next cue after the wait.

CURRENT CUE:

When RUN mode is off, the display shows the current cue rather than the main cue. This is the cue that the line pointer (>) is pointing at.

TASK 1 CUE, TASK 2 CUE, TASK 3 CUE:

These 3 lines show the current cue being executed in up to three tasks. As with MAIN CUE, if a WAIT is being executed, the next cue number will be displayed. If a particular time line is not active, a dash (-) is used.

MARK 1 CUE, MARK 2 CUE:

These two lines show what cues the current marks are set to, if any. A dash (-) indicates no mark has been set.

18. The AMPL/M Statements Menu has been updated to include UNTIL X, CLOCK PICKUP:, and ESCAPE. Since all the statements will not fit on a single screen, press <RETURN> to see the remaining statements. You can toggle back and forth between both statement lists with the <RETURN>.

19. The Editor Command Menu has been updated to include the Block Adjust command.

20. A new Multi-Image Utility has been added. This is called the Update utility. The purpose of this utility is to allow

you to update a AMPL/M System Master disk to operate with your serial number interface card. The update utility will not update to ANY OTHER serial number besides the one installed in the computer. Also, it will not update a disk of a different version number.

Complete directions for using this utility are given on the screen when you select the utility.

21. A new utility has been added to the Disk Utilities Menu. Press <F> to FORMAT DATA DISK. This utility will allow you to generate a data disk (no AMPL/M System) directly from AMPL/DOS. This is especially helpful if you run out of disk space while saving a file!

As usual, you are protected from errors. If the disk in the selected drive is not blank, you will be asked DISK ALREADY CONTAINS DATA: FORMAT (Y/N)?. If you wish to continue, press <Y>. If you blew it, press <N> or <ESC>.

The formatted data disk DOES NOT CONTAIN A COPY OF DOS 3.3! This gives you 32 extra sectors for data, 528 sectors in all, but the disk cannot be booted. If you try to boot an AMPL/M data disk generated by this utility, an appropriate message will appear on the screen.

22. The ACCESS MODE MISMATCH warning message is now operational. You will only see this message if you are in the Insert/Run, Edit/Run, or Single/Step modes. In normal RUN mode, the warning is NOT displayed, so it will not interfere with a running program in any way. For more information on this warning, and why it occurs, see the AMPL/M Reference Manual.

II. AMPL/M Bugs

As far as we know, all bugs in the system have been corrected. Some are too insignificant to mention, but the main bugs corrected are listed below, for your encouragement:

1. Syncrolink bug: Before this release, syncrolink would always operate on the main time line, unless you marked a block in a task. This was true even if you were viewing the task execution on the CRT screen (by autosyncing to the task). Now, you will only get syncrolink to work on the displayed time line, main or other, unless a block is marked. If the block is not displayed when it is executed (because you are looking at a different time line on the CRT), then syncrolink will NOT operate in the block either. This insures that syncrolink will not change something you can't see on the screen!
2. TASK delete bug: Before, if you deleted a task from your program, the task name would not be properly deleted in the task name data base, so the computer would think that the deleted task still existed, and that some other task didn't. No longer a problem.

3. Filename bug: Release 3.1.4 allowed you to type in filenames longer than 20 characters. This has been fixed.
4. Status computation bug: If you reversed from tray position 0 to 80, the tray position would remain at 0. This has been corrected.
5. Ship buffer overflow bug: Before, if you got a SHIP BUFFER OVERFLOW error, the program would keep running, even though you have missed some cues. Now, the system stops dead when this error occurs. This is also true of the NO TIME LINES AVAILABLE error.
6. Backup bug: Previously, if you made more than one backup copy at the same time, the second backup disk would not work. Now, you can make as many as you like.

III. Next Release Information

The next AMPL release, version 3.3, is scheduled for sometime this summer. Hopefully, it will include a new AMPL/M manual. If you wish to have some input for features to be included in release 3.3, please write a letter to Eric Anderson at Clear Light with your suggestions as soon as possible. All input will be appreciated.