



# General Purpose Software

1. **MIDI-Master Version 2.5Z** - A vastly improved MIDI program for those who already own the original version 2.3 or earlier. [Details](#)

## MIDI Master Version 2.5Z

This program package is the new and improved MIDI Master for those who do not have the AMS card in their TI-99/4A systems. It can also be used by Geneve owners, but is not recommended for that computer.

**NOTE WELL:** In order to use this program, you must already have the special adapter cable supplied with Versions 2.3 or earlier. This is an update. Although it has been subjected to major revisions in this version, this has been done with the express permission of the original author, Michael J. Maksimik, who still holds the copyright on this product.

Here are the major improvements made since Version 2.3:

Volumes can now be controlled voice-by-voice, so that you can make the drums less loud than the piano, the flute louder than the piano, and so on. In the old version you had only a single volume control, which affected all instruments together. Now, each individual track (16 are possible) has its own separate volume control byte, and can be set to any volume between 1 and 127 by entries in your SNF source file.

Duration of notes can now be entered in "tick" values (a quarter note equals 48 ticks) so that difficult timing situations (e.g. seven notes in the space of a quarter) can be handled easily without throwing the rest of the voices "out of synch" relative to this voice. This comes in very handy for pieces of jazz and other modern music, but even classical compositions have strange note timings on occasion.

All entry fields, such as those for file names, now have editing capability while in the field. Thus FCTN-1 deletes the character at the cursor, FCTN-2 starts an insert mode, and FCTN-3 clears out the entire field so you can start over if need be. The cursor blink rate in the entry fields is now controlled by the vertical interval timer, so that the blink rate remains the same pleasant speed regardless of whether the computer is a "stock" TI, a "bus modified" one, or a Geneve. (In Version 2.3 the blink rate was fine on Geneve, but painfully slow on a TI..)

The Pause function (press P while playing) now indicates by a blinking "PAUSED" on the screen when it's in effect. Also, when you engage Pause, the MIDI Keyboard is sent signals to make it

stop playing whatever notes were "in progress" when you hit the P key. In the old version, the MIDI instrument would be left with notes still "on and playing" while the program was paused. That could be very annoying indeed. The same is now true of the stopping of play through FCTN-9. The program tells the keyboard to "shut up" first, then comes out of play mode.

When a file has been loaded, either by the compiler or from a memory image file, the program keeps that name in a special place in memory. When the music is being played, there is an indication "NOW PLAYING" on the screen to let you know what file's contents you're listening to at that time.

The saving of files in memory image format has been greatly improved. In the old version, saving even a very short piece of music would create three files on your disk, totalling 100 sectors of disk space.

This was very wasteful and meant that the number of memory image pieces you could store on any one disk was limited unnecessarily. In this new version, the program determines just how much of the memory is actually occupied with the music, and saves only that much of the memory to disk. Thus a short piece of music may create only one file on the disk, of just the right sector size for that piece's content.

On loading memory image files, the program will "error trap" only if the first file of a series is not found. That was a necessary change because of the new, more efficient way of saving files in this version.

When running the compiler on an SNF source file, this new version will let you know if the program runs out of memory before finishing the compile. Previous versions would simply appear to continue compiling, but when the music played parts would "drop off" near the end of the piece with no apparent reason.

In the album mode (requires Mini-Memory, supercart, or other extra memory support) the random play mode plays each selected file once and only once but in random order. In the original version's album, random meant that play would continue until stopped by the user, with many re-plays.

2. **MIDI Play-In** - A new program that permits the user to attach his MIDI keyboard to the TI in both directions. Music played on the keyboard gets stored in the TI's memory, then can be played back, saved to disk, or used to create SNF source files for MIDI-Master [Details](#)



Here are the major Features of Play-In:

**NOTE WELL:** In order to use this program, you must already have the special adapter cable supplied with MIDI-Master Versions 2.3 or earlier.

The main feature of Play-In is that the user can connect a cable from the MIDI keyboard's MIDI Out connector to the MIDI In connector on the MIDI-Master cable adapter and then have music played on the instrument be recorded into the TI's memory.

While the music is being played, the user may select an option to have the TI's monitor speaker produce a metronome sound to help pace his playing. Depending on the make and model of keyboard in use, it may be necessary to have the received notes "echoed" back to the keyboard's MIDI In port in order to hear what's being played. That too is an optional feature provided in the playing-in mode of program operation.

When the user stops playing for two measures, the program concludes the input session. If an AMS card is in use, the program allows the user the option to play more of the piece into another part of the AMS memory. This way, different sections of a longer work may be played in one at a time, then played back straight through.

Once the musical content is in the memory, the music may be played back from memory into the keyboard to serve as a "quality check" on the original playing. Other things may be done with the music once it's in memory. It may be saved directly to disk in memory image form (NOT the same memory format as MIDI-Master memory image files) or it may be used within the program to create one or more SNF source files (D/V 80) for later editing and for use through the compiler function of MIDI Master.

This program is designed to work either with or without the AMS memory. If AMS memory is detected, it will automatically be measured for size, and that will be shown on the screen as what number of 24K memory "groups" are present. For a 256K memory, that will be ten groups. For a 1 Meg memory, 42 such groups will be reported. In operation of the program, the actions of the program will be "tailored" to the size of memory that's in use.

3. **Harrison's Word Processor** - An Assembly program for those who absolutely CAN'T STAND TI-Writer or any of its "clones". This one takes a completely different approach, modeled on the "professional" word processors of its day. No, it's no threat to the folks who make Word Perfect, but then it doesn't require a six-week course to use it, either. It's all been designed to make the job of creating documents easy, making them easy to read on-screen, while providing a wide range of options for printing. [Details](#)

## Harrison's Word Processor

Here are the major Features of the Word Processor:

The main features are ease of use and flexibility in interfacing with printers. This program was designed originally for our own use, so our "biases" show heavily in its design. All operations start with full-screen menus that require only the space bar and ENTER keys to select operations. Unlike TI-Writer and its "clones", this is a page-oriented WP, so that each page of a document is a separate file, and thus documents up to 126 pages in length can be created.

The text as displayed on-screen is always word-wrapped to the forty character screen, so there's no need to keep doing horizontal scrolling to read the document. The printing layout is determined separately, by user input when getting ready to print, but once that's done the desired layout is stored within the document as default settings.

**NOTE WELL:** This is an old program, and has NOT been updated for compatibility with hard drives. It will operate even if there's a hard drive present, but cannot be run from or save its documents on a hard drive of any kind. It is, however, completely compatible with Horizon Ramdisks, and that's the recommended way of using it, if available.

This program has been designed so that in many respects the user can "tailor make" many aspects of its operation to his own preferences, and it will "remember" all those preferences the next time it's started up. Perhaps the best example of this aspect is the matter of changing the printer actions when installing the program or when you've purchased a new printer. There are several built in printer action tables that can be installed in the program with the touch of a few keys, but the user can also create or edit a custom printer action table to match his own choice of printer. It's been used here at our house with three different models by Star Micronics and with a Canon Bubble-Jet (BJ 200e) one. With just a little care taken in setting up the Printer Action Tables, it worked perfectly with all of those.

Many other aspects of the program, ranging from screen color combinations through default printing setups (Margins, print strike, line spacing, etc.) can be changed and then saved so that they'll stay in effect on re-starting the program. The default disk drives to use for the program and its documents can also be set very easily and remembered.

Many handy-dandy features have been packed into this program. Just for example, there are two different kinds of insert modes, short and long. If you're doing a long insert and miss a character, you can use short insert within that long insert without having to come out of the long insert mode. There are also two kinds of delete, short and long. Short does what you'd expect, just deleting the character at the cursor position. Long delete marks the current character, then lets you move the cursor to the end of the section to be deleted and press the key (CTRL-1) again to remove all text between the cursor positions.

There is also a quick and easy way to embed control codes or escape sequences into the text by typing them as if in Extended Basic. For example, you could type `CHR$(27);"R";CHR$(5)` into this special entry mode (CTRL-3), and when you press CTRL-3 again, those three characters would be in place. The escape would show as a reversed video, as would the 5, with the R appearing as normal between them.

Moving and copying text have been made so that you don't have to move or copy in lines, but can move or copy text between any two cursor positions. Thus you could move or copy anything from just one word (or even just one letter, for that matter) to a whole page. The move and copy functions can move or copy text even from one document to another, even into one you haven't created before! We do that, for example, with our letterhead, which is a document of its own. We go into that, mark the whole thing to copy, then get out of that document, start a new letter to someone, and simply copy the letterhead in at the beginning.

There are many other features that we can't list even here in the "details" page. The manual for this program runs to fifty pages, on a separate disk (supplied) with its own Extended Basic program to print the manual.

4. **Font Designer** - A program for those who have either 24-pin or Bubble-Jet printers. Allows the user to create and download custom fonts to the printer. Disk includes several ready-made fonts that can be used as-is to "dress up" text printed by your TI. [Details](#)



Here are the major Features of the Font Designer:

The main features of this program disk are that it provides means to design and download custom fonts for 24-pin and Bubble-Jet printers. It's compatible with 24 pin dot matrix printers made by Epson, Star Micronics, and Panasonic. It's also compatible with Canon Bubble Jet printers in their Epson emulation mode. Most Panasonic 24 pin models will also have to be placed in their Epson mode. It may be compatible with other makes, but we don't know any for certain. This program disk WILL NOT WORK for 9-pin printers of any kind. The disk has three main programs, each of which is in E/A Option 5 format:

1. CHAR - The main designing program
2. QDUMP - A quick way to download
3. QTEST - A quick way to test what's been downloaded

In addition, there's an Extended Basic program called LOAD, which will load and run CHAR from Extended Basic. The E/A Option 5 programs can be loaded and run from E/A, from Funnelweb, or from a Ramdisk menu.

The CHAR program is the meat of this disk, because that's where the designing and re-designing of fonts takes place. There are eight items on its main menu:

1. LOAD CHARACTER SET
2. SAVE CHARACTER SET
3. EDIT A CHARACTER
4. DUMP TO PRINTER
5. DELETE A CHARACTER
6. RUN QUICK PRINT
7. PURGE FONT MEMORY
8. EXIT PROGRAM

The first one is a good way to start, because you can load in one of the seven fonts that are already on the program disk, have a look at its design, and even play around with modifying some of its characters.

Let's say for the moment that you have the disk in Drive 1. If you answer the INPUT FILE NAME prompt with DSK1. (including the period) the program will catalog the disk and show you

only the names of files that are suitable for use with this program. There will be seven of them in this case. Move the cursor beside one of the listed names and press ENTER. The program will load that font into its memory space, ready for editing. This process of disk cataloging will work also for Ramdisk drives and hard drives, including SCSI type. If, for example, you've got some of these font files on the SCSI1 drive in a sub-directory named FONTS, you'd enter SCSI1.FONTS. at the prompt (name MUST end with period) and would very quickly get the list of fonts in that directory on the screen.

Now that you've got a font in memory, you can use the other features. If your printer is connected and turned on, you can immediately download the font to the printer by choosing 4 from the main menu. A default file name of PIO.CR will be in place. If your printer is connected to that port, just press ENTER.

The font will be downloaded to the printer in a few seconds. When that's finished, your printer will be set up to print in the font you just downloaded. You can run a quick test print by selecting 6 from the menu. The printer will print all characters in the range that was included in the download, one after the other. This way you can see two things: First, what the font looks like, and second, what range of characters is covered.

Editing a font is done one character at a time. Press 3 off the main menu. Now press the character key for the character to edit. Try for example upper case A. The screen will show a grid of 24 by 36 small squares. The font dots for that A will be black on that grid. Because of the difference in aspect ratios between the screen and the printer, the letter will appear much wider than it does when printed, and it's difficult sometimes to picture what that array of fairly large black squares will look like on paper.

Pressing P will produce a smaller "preview" size of the character. Pressing the space bar will take that preview away.

Complete instructions are of course on the disk, but before we leave you we should briefly discuss those other two E/A Option 5 programs.

QDUMP is used when you have some font file that you want to download to the printer very quickly. It will, like the program CHAR, ask for a file name, and if the name you enter ends in a period, it will catalog the chosen path for you. When you select a font from that list, it will immediately download that font and make the printer ready to print in that style.

QTEST sends a short test printing to the printer, showing the font in normal printing, in italics, and in double-size characters, just to demo that particular font.

That's it! If you think this might be useful, it will cost you just \$1.00 to find out for sure. Try the disk yourself, and we think you'll like it.

5. **Word Search** - A program that takes lists of words supplied from D/V 80 text files and arranges them randomly into a matrix in eight directions. The result can be printed immediately or saved to an output D/V 80 file and printed later. A good number of ready-made word lists are included.

6. **Easy Data** - A disk of Extended Basic with embedded Assembly routines. This disk is designed to allow somebody with minimal knowledge of Extended Basic and absolutely no knowledge of Assembly to create and maintain small data bases of his own. Very quick and effective sorting by any two fields is done by an assembly sort that is embedded in the program. [Details](#)



Here are the major Features of Easy Data:

The main feature is its ability to permit the user to create his own small data bases without having any special knowledge, and without massive programming efforts.

This disk has a series of small programs in Extended Basic with Assembly routines already embedded in them. The user need not have more than a small amount of knowledge of Extended Basic, and needs no knowledge of Assembly whatsoever. The instructions provided on the disk, plus some ideas about what kind of data and how much, should be enough to let the user create a very easy to use data base.

**NOTE WELL:** This disk is NOT designed for keeping any "all world" data bases, as the limits of memory are still there, and all of the program and DATA has to be stored in the 24K of high memory expansion. A list of your close friends and close family would be fine, but not a city directory unless you live in a very small city.

One of the most valuable features in this program is its ability to sort the data on any field, with any other field as a secondary sort. In the sample program BIGDEMO, supplied on the disk, there are data records containing Last Name, First Name, street address, city, state, and zip code, each as a separate field. The user can choose any of those as the primary sort criterion, and any other one as secondary sort. Let's suppose you wanted to do the "normal" thing. You'd choose to sort by Last name as primary, then First name as secondary. This would create a sorted list where the records would appear in last-name order, but if two or more people had the same last name (as in a family data base) those with the same last name would be sorted in order by first names. To put that another way, if and only if there's a "tie" between two records on the primary sort does the program resort to the secondary sort to break the tie.

The user could choose to sort by state as primary and zip code as secondary, and that too would work, sorting everyone by state, and then within a state by zip codes. The user can make his own choices, as the program doesn't care.

All of the sorting process is done by an embedded Assembly routine, working from user-supplied DATA lines in the Extended Basic part of the program. Depending on the situation, the user can choose to have certain fields sort on a numeric basis instead of as strings, simply by assigning numeric variable names instead of string variable names to those fields. The user can

add to his data base at any time by just tacking on more DATA lines at the end of the XB program.

The disk contains two demo programs, a "skeleton" program which the user fishes out with his own data, and a couple of "merge" files that enhance the package. The whole thing comes on just one disk, with on-disk instructions, and all that's needed to use the stuff is a 32K memory, Extended Basic, and disk drive. Give this one a try, and you'll probably like it.

7. **TI-Artist Print** - This program prints black and white TI-Artist pictures. Unlike TI Artist, this program prints the pictures full size on 8 1/2 by 11 paper and in correct aspect ratio. Thus circles on screen are circular on the paper, squares are square, etc.

8. **TI-Artist Color Print** - For use with either 24-pin or Bubble Jet color printers. Like the black and white printing program, this preserves aspect ratio on the printout.

9. **Drawing Program** - A "poor man's" drawing program. Includes the ability to draw straight lines, circles, and combinations of these elements. Allows use of font files for adding alphanumeric data, annotations, etc. [Details](#)



Here are the major Features of The Drawing Program:

The main purpose of the Drawing Program is to have an alternative to the difficulties of using TI-Artist. It grew out of an experiment in the use of Bit-Map mode on the TI. The program includes the ability to draw lines and circles, to fill closed areas with color, to put alphanumeric stuff on the screen, and of course to erase mistakes.

Lines can be drawn by just putting the "pen" in its down state and moving the cursor in any of eight directions with either the arrow keys or with a joystick. This makes for a rather limited capability, so we also have the "line draw" mode. In that mode, the original cursor position serves as a starting point, and a second cursor moves with joystick or keys without leaving any line or moving the first cursor.

When this moving cursor is in the desired position, a press of the Q key or the joystick's fire button creates a straight line between the two cursor positions. From there, another line may be drawn in this same fashion. This makes it easy to create squares, triangles, or any other shape composed of a series of straight lines. Pressing the F key exits from this mode.

To draw a circle, one places the cursor where the center will be, presses the 5 key, moves the cursor up, down, left or right to a point that is on the desired circumference, then presses 5 again. The circle appears, and then the cursor returns to the center. This makes it very easy to create concentric circles or circles that have their centers along a horizontal or vertical line.



All sixteen colors can be used for any of the drawing functions, selected by simply pressing the 3 key. The color in use for drawing at any given time is shown by a small block in the lower right corner of the screen. That same color will be used for automatic filling of any enclosed area on the screen. To fill, one places the cursor inside the area to be filled, selects the fill color with the 3 key, then presses the 4 key.

To add alphanumeric data, such as labels or annotation, one simply presses the A key to put the program in its alphanumeric mode. Normally the characters that appear on the screen will be the regular TI screen characters, but the program's main menu has a selection for "load font", through which one can load either of two kinds of fonts. Fonts of the type used with TI-Artist can be loaded and used simply by using the complete file name including the \_F ending.

If you prefer, you can also load and use fonts of the CHARA1 type by simply using the file name of that file. (several of that type are included on the disk) In either case, the stuff put on your drawing in alphanumeric mode will appear in the font you've most recently loaded.

The foreground and background colors for the alphanumeric characters are selected by using the 1 and 2 keys. Again all sixteen colors are available. The alphanumerics get placed on a grid of 21 by 32 spots.

The resulting drawings can be saved in either of two formats. The default is a special format that puts both the black and white and color data in one 43 sector "program" type file. If you prefer to save in TI-Artist format, simply append \_P at the end of the file name. When you do that, the program will give you the choice of saving the color or not. If there's nothing but black and white, you can answer no to this prompt and the program will save only one 25 sector file with the \_P ending. If you answer yes, the program will also save a 25 five sector file with the same root name but ending in \_C. These two files will constitute a "normal" set of TI-Artist picture files that can be used by other programs.

The program also has an option on its main menu by which you can load TI-Artist Instance files for adding graphics to your drawing. Instances can be put on the drawing in two ways, either as an item that "cuts a hole" in anything else to make room for itself or as an item that simply overlays without taking away any existing items already on the drawing.

Finally, the program has a built-in printing function that will print your drawing in correct aspect ratio nearly filling an 8 1/2 by 11 sheet of paper. Correct aspect ratio means that if you've made a circle on the drawing, it will print as a circle on the paper. (TI-Artist doesn't seem to be able to do that.)

There are three different versions of this program, and the choice depends on the kind of printer you have. There is a "normal" version that works for 9-pin Epson compatible printers, a 24-pin version for those with more modern dot matrix models, and a Bubble Jet version for use with Canon Bubble-Jet printers.

**10. Assembly Poker** - A very fast and colorful five-card draw poker game between you and your TI-99/4A.

11. **Metronome** - For musicians. Use the TI's sound to provide a steady and accurate pacing for your practice session. Timing selectable in Beats Per Minute.

12. **Load Master V.2.2** - A collaboration with Mickey Cendrowski. This greatly improved version uses Assembly support routines to speed up operation and add more capabilities. [Details](#)

## Load Master V.2.2

Here are the major Features of Load Master V.2.2:

The main feature of Load Master V.2.2 is that it's a disk cataloging program which performs special identifications of file types. For example, it knows the differences between type 5 files that are Extended Basic programs and those that are Editor/Assembler Option 5 program files. It can also tell if an otherwise "program" file is actually a TI-Artist picture, and so on. There are over forty specific file types that get correctly recognized and reported as what they really are. This makes the listing on the screen make much more sense, in that there's a description that indicates the actual use for that file.

This program, which runs under Extended Basic, is the result of a collaboration between the Assembly Guru and Mickey Cendrowski of the West Penn Users' Group. Mickey originally created the program almost entirely in Extended Basic, with a few embedded Assembly routines from other Assembly authors. The program worked quite well, but was painfully slow. Having seen her demo on video tape from Lima (or maybe that was Cleveland) I offered to "pump in" more Assembly routines to speed up and improve the processing, particularly the business of making more complete identification of file types.

After about a year of collaboration, the final product was released as V.2.2, which is the product available today. It's a "Fairware" product, so donations for its use should be sent to Mickey Cendrowski. (address is on the program) Mickey shares those donations with me.

In addition to just cataloging files correctly, Load Master can also do other things. It can print out the catalog data just as a list, or it can create a jacket for the disk that has the catalog information printed right on the jacket. (Making the jacket requires some cutting, folding, and pasting, but the cut lines and fold lines are all printed right on the sheet, so it's easy to complete the job.)

While Load Master's catalog is on the screen, you can actually make use of the files right from the catalog listing. For example, you can "click on" a D/V 80 text file and read it right there without leaving Load Master. If it's an Extended Basic or E/A Option 5 Program file, it can be loaded and run directly from the catalog listing by again just "clicking" on the catalog entry.

Even E/A Option 3 object files can be loaded and run right from the catalog listing. In the cases of the E/A files, though, you must have at least two disk drives, since the Load Master disk has to be in its own drive (doesn't matter which one) so that it can load the E/A utilities into low memory before loading and running the E/A file. Putting those utilities into low memory "fools"

the TI into thinking it's running under Editor/Assembler even though the Extended Basic cartridge is still in place. I've used the word "click", which is a PC/Mac term in this part, by which is meant that you move the selection bar to the desired item and then press ENTER on the TI keyboard.

13. **Speed Read** - A program to improve your reading speed by displaying D/V 80 files of your choice at selected words per minute rates. You choose your own pace, then try to keep up with the computer.

14. **Code Breakers** - Solve cryptograms right there on your TI-99/4A screen. Includes several modes of operation, including a two-player challenge mode. [Details](#)



Here are the major Features of : Code Breakers

This is a program that allows you to solve cryptograms on the screen of your TI-99/4A. There are three important selections on the main menu:

1. Play from a File
2. Player Challenge Mode
3. Make/Edit Puzzles

Of these, number one is the primary choice for the lone player. It produces a prompt for the level of play, with choices of standard or professional. Standard play is highly recommended for the beginner or even a somewhat skilled solver. This level keeps the words and punctuation intact in the phrase to be de-crypted. Professional level is for the truly gifted player, in that it puts the content of the cryptogram into five-letter groups with no indication of word boundaries or punctuation. (Very Difficult!)

Next there's a prompt for the file name, with DSK1.PUZZLE as the default entry. That is the file of "wise sayings" that's included on the disk. There are a lot of ready-made cryptograms there. Actually, they are not encrypted in the file, but the program encrypts when it gets a particular one from the file. Each one retrieved gets encrypted with a randomly chosen encryption alphabet. During any one session of play, no record will be repeated. To solve, you just place the cursor on a letter in the cryptogram and hit the letter you think that one is. All occurrences in the cryptogram will be replaced with what you typed at that one place. If you get it messed up enough that you need to start over, just press FCTN-9 to put the whole thing back to its starting condition. Letters you've replaced show in a different color scheme so it's easy to see what you've changed and what's left to change. When you're done, press ENTER to see if you're correct.

Item 2 from the menu is for the case where two or more players are available. While the other player or players look away, a player types in something for the other to solve. When this first

player presses ENTER, the program encrypts that for the next player to solve. Two players can thus spend hours on end challenging each other.

Item 3 is there to allow the user to make his own files of puzzles or to edit ours or his own. **BE WARNED** that some of these puzzles are very difficult indeed, as they include many archaic words. They came mostly from a set of PC disks called "Wisdom Through the Ages", and thus many are in now obsolete English.

As usual, this disk contains complete instructions, and the programs can be run from E/A, from XB, from Funnelweb, or from a Ramdisk menu. It's being offered as Public Domain software.

# General Utility Software

Over the years since starting to use Assembly language, the Guru has produced a large body of work in various forms. Many are complete programs of one kind or another, but there are also some that fall into the category of "tools" for other programmers to use. Some are for use by Assembly programmers, some for Extended Basic programmers, and some for C99 programmers. Those listed below are not all of the utilities produced, but just some of the "highlights". All those listed have been released as Public Domain software, available for anyone's use without royalty or payment except to cover media and mailing.

---

1. **Extended Basic Utilities** - Volumes 1 and 2 are available. These are small enhancement routines to provide better and/or faster performance in Extended Basic programs. [Details](#)

## Extended Basic Utilities

Here are the major Features of the two volumes:

These two volumes contain Assembly routines for "embedding" into Extended Basic programs. Most are intended to improve speed of operations, but some are intended to perform operations that just plain can't be performed within Extended Basic. Some are pretty exotic. For example, volume one includes a routine to assign musical note frequency values to an array variable. That was by special request of the late Jim Peterson, and perhaps nobody else will ever use it, but it's there on the disk.

Volume one has one really valuable utility. That is the small (4 sectors) Object file called TRACK4. This performs the task of determining from which disk drive an XB program was loaded, then changing all references to DSK1 within the program to the drive or other device from which it was loaded. Thus an XB program that expects to open or run other files from Drive 1 can instead be run from Drive 2 or 3 or 8, and TRACK4 will have replaced all those DSK1s with the drive the program is in, thus avoiding the series of I/O errors which would otherwise result from running the program on a different drive. This works for any ordinary floppy disk or Horizon Ramdisk drive, and for the root directory of hard drives. Alas, there's no way for this routine to find out if the program loaded from a sub-directory on a hard drive, so its application to those is severely limited. It does work very nicely, though, for floppies and Ramdisks. Instructions for its use and a set of demo programs are included on the disk.

There are other small utilities on Volume 1, including special versions of Accept At type routines for strings and numeric variables. Haven't had to use those in quite a while myself, but the instructions are on the disk, as are demos of their operation. Get the disk and see for yourself.

Of all the utilities in Volume 2, the most valuable is the very fast menu driver. This routine uses DATA lines that the programmer puts into the XB part of the program, and uses that data to make menus appear instantly on the screen. This works so much faster than using READs and DISPLAY ATs that once you've seen the demo, you'll want to use it in any XB program that needs to display menus.

Other equally fast utilities are on that disk, mainly concerned with assigning either numerics or strings to array variables from DATA lines. These replace the old FOR - NEXT loops that read data into arrays, and perform the task about a thousand times as quickly. Like Volume 1, this volume has complete instructions to tell you how to use its routines with your own XB programs.

Both volumes include Todd Kaplan's ALSAVE and ALLOADM files to make it easy to "embed" assembly routines in your XB programs, and the instructions explain just how to go about using those utilities.

We do have other odds and ends of routines for use under Extended Basic, which I've not bothered to list on these pages. For example, we have routines that will allow an XB program to have music playing in the background during CALL KEY or ACCEPT AT operations. Quite exotic, but nice.

2. **MIDI Toolbox** - A whole series of Assembly programs for performing operations on the D/V 80 source files that are destined for use by MIDI-Master. These include a file combiner, a transposer, and even a measuring tool to determine whether an SNF file will fit in memory when compiled by MIDI-Master. [Details](#)



Here are the major Features of the MIDI Toolbox:

All of the tools in the toolbox are designed to work with D/V 80 files in the SNF source code format used as source for MIDI-Master. There are a total of six "tools", each of which is an Assembly program in E/A Option 5 form. The numbering system is inconsistent, in that there are no TOOL1 and TOOL6, and one of them is called simply COUNTER, not a "TOOL" file name. MY FAULT, but don't hold that against this valuable disk. Here we'll briefly describe each "tool" in the box. Complete instructions for their use are of course on the disk.

**TOOL2UT** - This is a tool to make it easier to handle relatively long compositions, or those with many tracks, by allowing a whole bunch of separate files to be combined into one large file for the MIDI-Master compiler. My partner uses this tool all the time, so that she can work on each track independently, then put them all together for compiling. The program is extremely easy to use.

**TOOL3UT** - Does the opposite of TOOL2UT, taking a large file and splitting it into pieces that are small enough to edit with Funnelweb, the E/A editor, or TI-Writer. This is a "dumb" file splitter, simply changing output files after a fixed number of lines have been processed.

**TOOL4UT** - A very intelligent tool, designed primarily for taking source files designed to work on TI and convert the "delay factors" for use with Geneve, or vice versa. It has other uses, though. The user can enter his own conversion factor, and thus change the tempo of a MIDI Master source file by any desired factor.

**TOOL5UT** - A "smarter" file splitter, which splits a large source file into multiple files track by track. Some musicians prefer to work that way, including my partner.

**TOOL7UT** - Remember, we said there's no tool 6. This tool, number 7, is a transposer. With this, the musician can change the key of the source file by adjusting all notes in the music up or down by a user-selected number of half-steps. This too is a smart tool, so that the user can, for example, change just one track or even part of a track by a whole octave or more. This was recently updated so that it can handle notes one octave below MIDI-Master's C0 and up to C7.

**COUNTER** - This is a "predictor" program. It takes a MIDI source file and determines in advance whether that file will fit in memory, and if not, by how much it will miss. In the instructions for this tool is a description of the mapping of memory use by MIDI-Master, which should take some of the mystery out of the program.

3. **File Converter** - A very special little program to take downloaded text files that are in D/F 128 format and convert them into D/V 80 files for editing.

4. **ScsiCat** - Special cataloging program for users with SCSI hard drives. Also works for regular floppy disks and Horizon Ramdisks.

5. **Randoms** - A utility disk for Extended Basic programmers and others who want very quick and very nicely randomized numbers.

6. **ASM Sound** - Utilities for Assembly programmers to make it easier and quicker to incorporate sound list music of effects into their programs. [Details](#)

## Sound Utilities

Here are the major Features of our "sound" utilities:

There are two disks covered by this description, one for use by Assembly programmers, the other for those who prefer C99 programming. Except for the "interfacing" to their respective main programs, these perform in exactly the same way, so we're saving on space by treating both in a single "details" article.

The routines on these two disks deal with what TI calls "sound lists", which are collections of data that can be sent to the sound chip via an internal interrupt, thus allowing the sound to proceed "on background" while other things are happening in the main program. This can be a very useful feature, limited though the sound chip may be. The new PC that I'm using to create this stuff has its own version of that idea, in that a regular music CD can be placed in the CD Rom drive and it will then play on background while I'm working here at the keyboard. What we're offering for the TI isn't nearly that good, but it does work and can create some really nice effects if you use it properly.

The following discussion assumes that you have prepared one or more sound lists in advance to incorporate them into your program as data. The first utility, then is one that will take up to ten separate sound lists from within your program's data and load them into VDP Ram. This process is fairly simply done, and the instructions supplied on the disk tell you exactly how to use this feature. By the way, we also have a utility listed on the Utilities Software page that makes it easier to prepare sound lists of the musical variety from sheet music. That's called Sound List Maker.

Once that loading process is done, any of those lists can be activated by your program at any time by using the Start Sound routine. This routine is really the heart of the utility, because it makes everything else happen. There are four different ways of starting the play of a particular sound list that you included in the loading to VDP ram.

- TYPE 1 - Play continuously on background
- TYPE 2 - Play Once, then stop
- TYPE 3 - Play Once, interrupting type 1
- TYPE 4 - Play once, delaying the main program

Except for Type 4, all of these make the sound play "on background", so that the main program may be doing something else, such as awaiting a keystroke. The Type 3 sound is perhaps the most interesting. If a Type 1 (perhaps background music) is ongoing, you can activate a Type 3 (perhaps a sound effect) which will interrupt the type 1 just long enough to finish itself, and then the Type 1 will pick up right where it left off.

The Type 1 itself need not be hundreds of measures of music. When a Type 1 has been activated and comes to the end of its sound list, it will simply "da capo" all by itself. In other words, when it ends it just starts over "from the top".

Of course there will be times when your program needs to stop the sound, and there's a routine that will do just that, stopping whatever sound is playing (except Type 4 for obvious reasons) but will leave all the sound lists in place in VDP Ram so they can be re-activated without re-loading them. There's also a routine that will stop the sound and eliminate the sound lists from access. This may be necessary in cases where your program needs to do disk access and that would corrupt the sound lists by using VDP Ram as a buffer for the disk input.

And finally, there are even routines that will allow some action in the main program to be synchronized to the notes in background music. That is, for example, having a cursor or other symbol on the screen move one space each time a new note starts in the music. It's a bit tricky to use this one, but it can be done if care is taken.



There are demo programs on these disks that illustrate the use of all the features we've mentioned here, plus the source code for the demos and the routines themselves, so that you can delve more deeply at your choosing.

7. **C99 Sound** - Similar set of routines for handling sound lists, but designed for use within programs written in C99. [Details](#)

## Sound Utilities

Here are the major Features of our "sound" utilities:

There are two disks covered by this description, one for use by Assembly programmers, the other for those who prefer C99 programming. Except for the "interfacing" to their respective main programs, these perform in exactly the same way, so we're saving on space by treating both in a single "details" article.

The routines on these two disks deal with what TI calls "sound lists", which are collections of data that can be sent to the sound chip via an internal interrupt, thus allowing the sound to proceed "on background" while other things are happening in the main program. This can be a very useful feature, limited though the sound chip may be. The new PC that I'm using to create this stuff has its own version of that idea, in that a regular music CD can be placed in the CD Rom drive and it will then play on background while I'm working here at the keyboard. What we're offering for the TI isn't nearly that good, but it does work and can create some really nice effects if you use it properly.

The following discussion assumes that you have prepared one or more sound lists in advance to incorporate them into your program as data. The first utility, then is one that will take up to ten separate sound lists from within your program's data and load them into VDP Ram. This process is fairly simply done, and the instructions supplied on the disk tell you exactly how to use this feature. By the way, we also have a utility listed on the Utilities Software page that makes it easier to prepare sound lists of the musical variety from sheet music. That's called Sound List Maker.

Once that loading process is done, any of those lists can be activated by your program at any time by using the Start Sound routine. This routine is really the heart of the utility, because it makes everything else happen. There are four different ways of starting the play of a particular sound list that you included in the loading to VDP ram.

- TYPE 1 - Play continuously on background
- TYPE 2 - Play Once, then stop
- TYPE 3 - Play Once, interrupting type 1
- TYPE 4 - Play once, delaying the main program

Except for Type 4, all of these make the sound play "on background", so that the main program may be doing something else, such as awaiting a keystroke. The Type 3 sound is perhaps the most interesting. If a Type 1 (perhaps background music) is ongoing, you can activate a Type 3 (perhaps a sound effect) which will interrupt the type 1 just long enough to finish itself, and then the Type 1 will pick up right where it left off.

The Type 1 itself need not be hundreds of measures of music. When a Type 1 has been activated and comes to the end of its sound list, it will simply "da capo" all by itself. In other words, when it ends it just starts over "from the top".

Of course there will be times when your program needs to stop the sound, and there's a routine that will do just that, stopping whatever sound is playing (except Type 4 for obvious reasons) but will leave all the sound lists in place in VDP Ram so they can be re-activated without re-loading them. There's also a routine that will stop the sound and eliminate the sound lists from access. This may be necessary in cases where your program needs to do disk access and that would corrupt the sound lists by using VDP Ram as a buffer for the disk input.

And finally, there are even routines that will allow some action in the main program to be synchronized to the notes in background music. That is, for example, having a cursor or other symbol on the screen move one space each time a new note starts in the music. It's a bit tricky to use this one, but it can be done if care is taken.

There are demo programs on these disks that illustrate the use of all the features we've mentioned here, plus the source code for the demos and the routines themselves, so that you can delve more deeply at your choosing.

8. **Sound List Maker** - A means to make sound lists more easily from sheet music. Allows the user to write in a quasi-musical notation and have the Assembler translate into Hex for you. [Details](#)



Here are the major Features of Our Sound List Maker:

The main feature is that with this software package and some sheet music, even a person with very little musical talent (like the Assembly Guru) can actually make musical pieces into the proper instructions for the sound chip so that pieces can be played "on background" using the interrupt cycle of the TI-99/4A.

The key to the operation of this "list maker" is in a simple assembly source file which is provided on the disk. That source file contains no data and no instructions, and won't occupy any space in memory when assembled. The whole file is nothing but a long series of EQU directives. These EQU directives establish definitions for special names that you use in creating the source file for your musical piece. You write in a different language in that music source, and the Assembler uses those equates to translate what you write into all that Hex stuff which eventually gets fed to the sound chip.

Let's simply show one example, without getting too technical. You can write this into your music source file:

```
BYTE 11,G1+G4SA,G4SB,G2+E5FA,E5FB,G3+A1SA,A1SB  
BYTE G1V+3,G2V+5,G3V+7,G4+6,G4V+7,E
```

This makes generator 1 play a 4th octave G sharp, generator 2 play a 5th octave E flat, and generator 3 play a 1st octave A sharp. That second line makes the volume 3 on generator 1, 5 on generator 2, and 7 on generator 3. It also makes the noise generator play a note three octaves down from what generator 3 is playing, at volume 7.

That's probably not good music, and it's probably not worth trying, but it's just there to illustrate that no Hex has to be understood to make sound lists with this utility. It's very hard to make something like this understood without actually trying it yourself, but trust us, it can be done quite easily.

On the disk are actual pieces of music that were done this way, in demo programs that illustrate the fact that it works the way we say it does. This disk is mainly intended for those who have just a little knowledge of music and almost no knowledge of how to make sound lists for Assembly or C99 programs, but do have knowledge of one of those two languages. If any of this interests you, get the disks and try it.

9. **C99 Utilities** - A collection of handy-dandy Assembly modules to make the life of the C99 programmer a bit easier. [Details](#)



Here are the major Features of the C99 Utilities:

The main features of this package are in eight object modules designed to add "new and improved" function calls for C99 programmers. There are multiple functions in most of these modules. It was split up into modules so that only those functions that were absolutely necessary in any one program would need to be included.

All of these were developed because of our friend Vern Jensen. He wrote and asked for these assembly routines to make some things easier in his C99 programming efforts. Over the course of about a year, Vern and I collaborated on these modules until we got them to what he and I thought was "exactly right". Here are those module names:

1. ACCNUM/O
2. ACCSTR/O
3. DISSTR/O
4. PUTNUM/O
5. PUTNUM2/O
6. SEGSTR/O
7. SNNS/O

## 8. KEYJOY/O

These names are of course short. We'll now briefly describe what each does.

ACCNUM/O is a kind of ACCEPT AT for numbers. The entry field is just six characters, because only integers are to be accepted.

ACCSTR/O is, as the name implies, an ACCEPT AT for strings. In this case the field length is determined by a parameter supplied by the C99 program.

In both of the above, as you'd expect, the C99 program supplies the desired row and column for the start of the input field, and in both there's a parameter that determines whether or not to clear the field before taking input. Both have the editing capabilities on FCTN-1, FCTN-2, and FCTN-3 just as in Extended Basic's case.

DISSTR/O acts as a DISPLAY AT for strings. The C99 program provides the row and column parameters, etc.

PUTNUM/O places an integer number on the screen at a selected row and column. The C99 program specifies whether the number is to be left or right justified in the six character field. The display field gets cleared before the number is placed there.

PUTNUM2/O works the same, except that it doesn't clear the field.

In all of these routines, the routine can be used when operating in either 32 or 40 column mode, and the row, column parameters will be interpreted correctly, because the routine determines by itself which screen mode is being used, and behaves accordingly.

SEGSTR/O takes a segment of one string into another, much like the SEG\$ function in Extended Basic, except of course that both input and output are ASCIZ strings, in keeping with C99's practice.

SNNS/O performs either of two functions. It either converts an integer number into a string or converts a string into an integer number, depending which of its two functions is called.

KEYJOY/O is a multi-purpose module for dealing with keyboard and joystick inputs. It performs functions like CALL KEY and CALL JOYST, but offers additional features such as a "button" call, which reports whether or not either joystick's fire button is being pressed. This is a sophisticated and powerful little module.

Those who have purchased Vern Jensen's PC99 starter kit already have this disk, as he includes it as part of that package. In either case, whether you got it from Vern or get it from us, the disk and all its contents are in the Public Domain.

**10. Bit Map Modules** - Another little tool for the harried Assembly programmer who's trying to "turn the corner" into the Bit Map modes of operation. This disk has four object modules that can be incorporated into your own programs to provide easy transitions into and out of half bit

map and full bit map modes, PLUS automatic motion for sprites in both of those modes. (Impossible!) [Details](#)

## Bit Map Modules

Here are the major Features of the Bit Map Modules:

The main features of this disk are object modules designed to make it easy for Assembly programmers to use either full bit-map or half bit-map modes in their own programs. Using these modules will allow a program to switch to one of these modes, operate in that mode complete with sound effects, automatic sprite motion, etc. There are four main object modules, two for the half bit map mode and two for the full bit map mode.

- HBSUB/O
- HBSPR/O
- FBSUB/O
- FBSPR/O

Let's begin with the first one, HBSUB/O. First, of course, it provides subroutines called SETHB and SETGM. These set the system into Half Bit Map and normal Graphics modes, respectively. Thus your program, which would start in the Graphics mode by default, can gracefully make the transition into half bit map by simply performing a BLWP @SETHB.

Before your program exits, it can set things back to the "normal" situation by using the instruction BLWP @SETGM, thus going back to graphics mode. In between, it can operate in Half Bit Map using the other subroutines in HBSUB/O. Using those, it can clear the screen, display strings, display individual characters, and set the color schemes on a character by character basis.

If you want to use sprites while in half bit map, complete with automatic motion, you would also include the second module, HBSPR/O, which offers a full range of operations on sprites, including some you've never heard of, because we invented them. For example, there's one called REVM0, which reverses the motion of a sprite in both vertical and horizontal directions. There's one called REV1, which reverses the character pattern of a sprite left-to-right, and one called REV4, which does that for sprites composed of four characters.

The standard operations on sprites are all covered, including MOTION, LOCATE, PATTERN COLOR, COINC, etc. In other words, you'll have complete control over what your sprites do, and 32 of them to play around with.

Moving along now, for Full Bit Map there are the modules FBSUB and FBSPR. These perform essentially the same operations, but for the full bit map mode. The operations on sprites are the same. There are additional string and character display options in full bit map mode, plus a couple of different routines to plot individual pixels on the bit map screen.

There are complete instructions on the disk, and an XB program to print those out for you. There are also demo programs that illustrate the use of the routines, and one extra one that

illustrates doing several "impossible" things all at once in Bit Map mode. This particular demo uses just about all the routines in FBSUBS and FBSPR, plus the ASM Sound routines. When it runs, Sprites move automatically (impossible) and music plays in the background (also impossible) interrupted now and then by sound effects. The demo program MUSIC is, all by itself, worth the \$1.00 we ask for this disk.

11. **Reformat** - Takes D/V 80 files of the type made with TI-Writer and formats them into different column widths, right justifies if desired, and so on.

12. **Sandwich** - Two small pieces of Assembly that can be used with an E/A object file to convert that into a stand-alone E/A Option 5 program file, with or without having the original source.

13. **Instance Editor** - Permits creating and editing instance files for use with TI-Artist, etc. [Details](#)



Here are the major Features of The Instance Editor:

The main feature of the Instance Editor is the ability to load in existing Instance files as used with TI-Artist and other such programs, edit them in Bitmap fashion, and save them back to disk as edited. Instead of that, you can load in an existing TI-Artist picture file and take excerpts of that to save as Instances. You can also just start with a blank bit-map screen and create your own instance from scratch.

While you're in the edit mode, you can take advantage of a number of useful functions, including line drawing, making circles, and using alphanumeric annotations on your instances. Lines can be drawn by just putting the "pen" in its down state and moving the cursor in any of eight directions with either the arrow keys or with a joystick. This makes for a rather limited capability, so we also have the "line draw" mode. In that mode, the original cursor position serves as a starting point, and a second cursor moves with joystick or keys without leaving any line or moving the first cursor.

When this moving cursor is in the desired position, a press of the Q key or the joystick's fire button creates a straight line between the two cursor positions. From there, another line may be drawn in this same fashion. This makes it easy to create squares, triangles, or any other shape composed of a series of straight lines. Pressing the F key exits from this mode.

To draw a circle, one places the cursor where the center will be, presses the 5 key, moves the cursor up, down, left or right to a point that is on the desired circumference, then presses 5 again. The circle appears, and then the cursor returns to the center. This makes it very easy to create concentric circles or circles that have their centers along a horizontal or vertical line.

To add alphanumeric data, such as labels or annotation, one simply presses the A key to put the program in its alphanumeric mode. Normally the characters that appear on the screen will be the regular TI screen characters, but the program's main menu has a selection for "load

font", through which one can load either of two kinds of fonts. Fonts of the type used with TI-Artist can be loaded and used simply by using the complete file name including the \_F ending.

If you prefer, you can also load and use fonts of the CHARA1 type by simply using the file name of that file. In either case, the stuff put on your drawing in alphanumeric mode will appear in the font you've most recently loaded.

Also on the disk are small programs that will print instances via either 9-pin or 24-pin dot matrix printers. The 24-pin version will also work on Bubble-Jet printers. The printing programs will auto-center the instance on the paper.

As with most of our programs, complete instructions are included on the disk so that you can study the ins and outs of the program whenever you wish.

14. **Password** - For those with Horizon Ramdisks. Keep the kids out of your TI by having a secret password required to get the Ramdisk menu to appear.

# Super AMS Card Software

Please NOTE: All of the software items listed here are designed to make use of the AMS for their data storage. None of these uses, requires, nor can be used with the "system loader" provided with the AMS package. Each is an Editor/Assembler Option-5 program file. Each disk also has a LOAD program so that the Option-5 can be loaded/run from Extended Basic. Each program listed measures the size of the AMS card in use, and tailors its operation according to the amount of AMS available.

**AMS Slide Show** - Shows TI-Artist pictures in sequence. [Details](#)

## AMS Slideshow

Here are the major Features of AMS Slideshow:

The main feature of this program is its ability to take a multitude of TI-Artist Picture files into memory at one time and then show them on-screen in rapid succession. The latest version has been designed for compatibility with SCSI hard drives, older type hard drives, and Horizon Ramdisk drives, as well as with regular floppy disk drives of all kinds.

The program on startup (after its title screen) simply asks for a path name in which to look for TI-Artist Picture files. The path name must end with a period(.), but may include levels of sub-directories. Valid path names include DSK1. SCS1.PICTURES. WDS1.TIPIX. and so forth, any path name being valid just so long as it ends with a period.

The program then does a selective cataloging of that path, reporting only those files that match the profile of a TI-Artist Picture file. (i.e. 25 Sectors length, "Program" type, and file name ending with \_P.

The list of such files may be up to 127 such files. The screen can be paged if necessary to view all the file names. Once the directory cataloging is finished, a selection cursor appears on the screen next to one of the file names. To select a particular picture for inclusion in the slide show, simply put the cursor beside its name and press ENTER. As you select, the number of items selected appears in the upper right corner of the screen. If you want an item to appear more than once in the sequence, that, too, can be done by simply selecting it again. Any one picture may be included up to three times in any one sequence. The order in which you select the items will be the order in which they're shown.



When done selecting, you press FCTN-6 to get the showing started. The program offers you the choice of having an automatic time delay before changing slides (from 0 through 300 seconds, which is five minutes) or to simply have each slide remain on-screen until a key is pressed. If the time delay method is used, time entered in the delay field is taken as a floating point number, so time may be entered as, for example, 1.25 seconds, 3.425 seconds, and so forth. Leaving the time field blank will mean a zero delay, in which case the computer will change slides as rapidly as possible.

The program will prompt for how many times to repeat the sequence. Leaving the entry blank means just one showing. Otherwise you may put in any number from 1 through 32767, and that's how many times the sequence will repeat. Before the first showing, all the selected pictures will be loaded into memory, and this takes some time, particularly if a floppy disk is the source. Once they're all loaded, however, showing will proceed rapidly or less rapidly depending on the choices you've made, but without need to access the disk drive again. There's even a pause feature, in case you need it. Pressing the space bar during the showing will "stop the clock" as long as the space bar is held down, then continue timing when the space bar is released. If you decide to stop the show before it's completed, you can press FCTN-9 to escape back to the file selection screen.

When it starts, this program measures the size of your AMS memory, and tailors its operation to the size present. Thus when you're selecting files for showing, the program will automatically limit the number you can select according to the memory size present. For a 256 K card, no more than 20 pictures will be allowed. For a 1 Meg card, 84 pictures will be allowed.

The program will determine all by itself whether any particular picture in the selected sequence has or does not have a color file associated with it. If it does, it will of course be shown in full color. If there's no corresponding file ending in \_C, that picture will be displayed in glorious black and white. A modest collection of TI-Artist pictures is included on the disk so you can try it right away.

**AMS Video Titler** - Uses TI-Artist or Harrison type pictures with special wipes for creating title sequences to put on VCR productions. [Details](#)



Here are the major Features of AMS Video Titler:

The main feature of this program is its many kinds of "wipe" transitions between the frames of video stored in memory. The purpose of the program is to provide video from the TI to a VCR or other video recording device to record a sequence of titles with various kinds of transitions between the frames.

There are twenty-three different transition "wipes" available, ranging from a simple top-to-bottom wiping through exotic effects like "venetian blind" transitions. Wipes can be made from corner to corner in four directions, from side to side, bottom to top, middle to edges either vertically or horizontally, etc. There are even two "random" wipes, in which pieces of one

frame replace pieces of the other in random order. With all these wipes to choose from, you can make the title sequences on your videos a treat to watch.

This program can use two different kinds of files as the source of frames in its sequences. You can use TI-Artist pictures or pictures made with our Drawing Program. The two types can even be mixed in a particular sequence without ill effects. The program will catalog the selected disk and show you the names of suitable files on the screen, from which you can select a file to include in your sequence.

Once the files are loaded, you can start the sequence at any frame which has been loaded with a picture, and then make your transitions when you choose by just pressing a key on the keyboard. The particular key you press determines which wipe happens. Each frame, once wiped onto the screen, stays in place until you press a key to wipe the next frame onto the screen.

The program has been made so that it's compatible with all types of disk drives, from the simple floppy type through Ramdisks, and even the SCSI hard drives. The disk includes seven ready-made picture files so that you can try it out right away.

Here are those wipe patterns and the keys to press:

- Rapid top to bottom (S)
- Rapid bottom to top (D)
- Slow top to bottom (E)
- Slow bottom to top (X)
- Slow left to right (A)
- Slow right to left (F)
- Center to top and bottom vertical (V)
- Top and bottom to center vertical (C)
- Center to sides horizontal (H)
- Sides to center horizontal (J)
- Spiral inward from edges (I)
- Spiral outward to edges (O)
- Downward Venetian Blind (Y)
- Upward Venetian Blind (U)
- Corner upper left to lower right (1)
- Corner upper right to lower left (+ or =)
- Corner lower left to upper right (Z)
- Corner lower right to upper left (> or .)
- Rapid random blocks (T)
- Slower random blocks (R)
- Through Green (G)
- Through White (W)
- Through Black (B)

These last three make the transition by having the screen go to the stated color for a moment, then the next frame appears instantly.

**AMS File Transfer** - Takes large text files via RS-232 from another computer (PC) and produces editable D/V 80 files on the TI. [Details](#)



Here are the major Features of AMS Transfer:

The main feature of this program is that it takes large text files into the AMS Memory and then creates a series of output files sized to be capable of editing using the most popular text file editors for the TI. Those include Funnelweb, TI-Writer, and other TI-Writer "clones".

The default source for such files on input is the RS-232 port at 1200 baud. The original intent of the program's design was to have the RS-232 connected by a straight cable (no Modems) to a suitable port on another computer, usually a PC of some kind. For this to work that way, one must have some kind of terminal program running on the PC. On our old Tandy 1000 SX, for example, we've successfully used its Deskmate program and Telix. Both of those allow selection of the rate and use of 7 bit with odd parity, which matches the "normal" input file setup for this program. The program will also work with large files that are already on TI disk sources, just by replacing the default input file prompt with the appropriate file name. (e.g. DSK2.BIGFILE)

Many of the text files that people have on their PCs are downloads that may not be correctly set up to transfer smoothly into D/V 80. Often these files use a "new line" character, actually a line feed [CHR\$(10)] to signify in the middle of a record that a CR/LF should be performed. In this program, we use that line feed to split an incoming record where it's found into two records in the D/V 80 output file. At the split point, the LF gets replaced by a CR, and the remainder of that incoming record starts a new record in the D/V 80 output file. This splitting happens each time the LF is found, so one incoming record may result in two or more records in the output file. This re-formatting where LFs are found makes the resulting files much easier to use and edit on the TI editing programs, such as Funnelweb or TI-Writer.

The process of splitting at Line Feeds also changes an already existing CR character into a space. In some files, particularly those created on the TI, it's necessary to have those CRs remain as is in the output files. As of October 15th 1998 the program has been updated so that the user has the option of turning this CR filter on or off. It defaults to ON, but this way the user has the option of leaving CRs as-is.

This program will operate with or without the AMS card, but if there's only a 32K memory, the size of files that can be handled is limited to 24K bytes. With a 256K AMS, files of ten times that size can be input. With a 1 Meg AMS, files of about a megabyte can be input.

There is another possible use for this program, involving cases where there are very large text files already on a TI disk drive. In such cases, the user can simply replace the default entry in the "Input File" field with the name of the large file on disk (e.g. DSK2.BIGFILE).

In any case, once the file is finished being loaded, there's a prompt for the first output file name. Since this will be the first of a series of files, the name should end with 1 or A, as the sequence of output files will be formed by incrementing the last character of the file name.

When the program finishes making the required number of (D/V 80) output files to contain all of the input text, it will simply go back to the input prompt in case you have more files to process. This program, like others of our AMS programs, measures the size of the AMS and tailors its operation to the size of memory found.

Because it's a small program that may have very wide use, we have released this program as Public Domain, so that it may be obtained, used, and copied without our permission and without payment except to cover media and mailing.

**AMS MIDI-Master** - Vastly improved version from the original. Uses AMS to allow much larger musical works to be played by your TI. [Details](#)

## MIDI Master Version 2.5B

This program package is the new and improved MIDI Master for those who have the AMS card in their TI-99/4A systems. It can not be used by Geneve owners.

**NOTE WELL:** In order to use this program, you must already have the special adapter cable supplied with Versions 2.3 or earlier. This is an update. Although it has been subjected to major revisions in this version, this has been done with the express permission of the original author, Michael J. Maksimik, who still holds the copyright on this product.

Here are the major improvements made since Version 2.3:

Volumes can now be controlled voice-by-voice, so that you can make the drums less loud than the piano, the flute louder than the piano, and so on. In the old version you had only a single volume control, which affected all instruments together. Now, each individual track (16 are possible) has its own separate volume control byte, and can be set to any volume between 1 and 127 by entries in your SNF source file.

Duration of notes can now be entered in "tick" values (a quarter note equals 48 ticks) so that difficult timing situations (e.g. seven notes in the space of a quarter) can be handled easily without throwing the rest of the voices "out of synch" relative to this voice. This comes in very handy for pieces of jazz and other modern music, but even classical compositions have strange note timings on occasion.

All entry fields, such as those for file names, now have editing capability while in the field. Thus FCTN-1 deletes the character at the cursor, FCTN-2 starts an insert mode, and FCTN-3 clears out the entire field so you can start over if need be. The cursor blink rate in the entry fields is now controlled by the vertical interval timer, so that the blink rate remains the same pleasant

speed regardless of whether the computer is a "stock" TI, a "bus modified" one, or a Geneve. (In Version 2.3 the blink rate was fine on Geneve, but painfully slow on a TI..)

The Pause function (press P while playing) now indicates by a blinking "PAUSED" on the screen when it's in effect. Also, when you engage Pause, the MIDI Keyboard is sent signals to make it stop playing whatever notes were "in progress" when you hit the P key. In the old version, the MIDI instrument would be left with notes still "on and playing" while the program was paused. That could be very annoying indeed. The same is now true of the stopping of play through FCTN-9. The program tells the keyboard to "shut up" first, then comes out of play mode.

When a file has been loaded, either by the compiler or from a memory image file, the program keeps that name in a special place in memory. When the music is being played, there is an indication "NOW PLAYING" on the screen to let you know what file's contents you're listening to at that time.

The saving of files in memory image format has been greatly improved. In the old version, saving even a very short piece of music would create three files on your disk, totalling 100 sectors of disk space. This was very wasteful and meant that the number of memory image pieces you could store on any one disk was limited unnecessarily. In this new version, the program determines just how much of the memory is actually occupied with the music, and saves only that much of the memory to disk. Thus a short piece of music may create only one file on the disk, of just the right sector size for that piece's content.

On loading memory image files, the program will "error trap" only if the first file of a series is not found. That was a necessary change because of the new, more efficient way of saving files in this version.

When running the compiler on an SNF source file, this new version will let you know if the program runs out of memory before finishing the compile. Previous versions would simply appear to continue compiling, but when the music played parts would "drop off" near the end of the piece with no apparent reason.

In the album mode the random play mode plays each selected file once and only once but in random order. In the original version's album, random meant that play would continue until stopped by the user, with many re-plays.

With the AMS in use, much larger pieces of music can be handled. With a "standard" 256K AMS memory card, ten times the music can be kept in memory at one time and played straight through. The AMS version also has the album included, but uses a normally unused part of the AMS memory to contain the code for album, and so does not need Mini-Memory. The AMS version WILL NOT RUN without an AMS available.

**MIDI Play-In** - Allows you to play music on the MIDI keyboard and have that music "recorded" in the AMS memory, played back, stored on disk, and used to create SNF source files for MIDI-Master [Details](#)



Here are the major Features of Play-In:

The main feature of Play-In is that the user can connect a cable from the MIDI keyboard's MIDI Out connector to the MIDI In connector on the MIDI-Master cable adapter and then have music played on the instrument be recorded into the TI's memory.

While the music is being played, the user may select an option to have the TI's monitor speaker produce a metronome sound to help pace his playing. Depending on the make and model of keyboard in use, it may be necessary to have the received notes "echoed" back to the keyboard's MIDI In port in order to hear what's being played. That too is an optional feature provided in the playing-in mode of program operation

When the user stops playing for two measures, the program concludes the input session. If an AMS card is in use, the program allows the user the option to play more of the piece into another part of the AMS memory. This way, different sections of a longer work may be played in one at a time, then played back straight through.

Once the musical content is in the memory, the music may be played back from memory into the keyboard to serve as a "quality check" on the original playing. Other things may be done with the music once it's in memory. It may be saved directly to disk in memory image form (NOT the same memory format as MIDI-Master memory image files) or it may be used within the program to create one or more SNF source files (D/V 80) for later editing and for use through the compiler function of MIDI Master.

This program is designed to work either with or without the AMS memory. If AMS memory is detected, it will automatically be measured for size, and that will be shown on the screen as what number of 24K memory "groups" are present. For a 256K memory, that will be ten groups. For a 1 Meg memory, 42 such groups will be reported. In operation of the program, the actions of the program will be "tailored" to the size of memory that's in use.

---

Bruce Harrison, who was the author of the programs and their explanations in this catalog, died on November 9, 2007. This catalog is a compilation of the information that Bruce put on his web site <http://members.aol.com/rottencat1/>, saved here so that it is not lost in the shuffle of time.

Bill Gaskill  
December 22, 2007