

GRAM Packer

Distributed by Genial Computerware

© 1986 J. Peter Moddie

GRAM Packer

Table of Contents

Overview	2
Loading the Program	2
Using the Program	2
Characters	3
DSR	4
Filename	4
Menu	4
Pack	4
Quit	4
Subprogram	4
TI Save	5
Saving Packed Files	6
Memory Usage	6
High Address Used	7
Testing a Pack	7
Trouble Shooting	8
Utility Files	8
An Example	11
Multiple Menus	11
Advanced Applications	
TI Save Quirk	12
Creating Utility Files	12
File Naming Conventions	13
GRAM Utilities	13
Suggestions on RAM Disks	14
Random Notes	14

Should you have any questions, problems or suggestions with this program, please write to us. We would like to here from you. If your copy of the GRAM PACKER is defective, you may return it within 90 days for a free replacement. GRAM PACKER is copy written but not copy protected. We do advise you to back up your copy to avoid paying a fee of \$3.00 to cover our costs.

Genial Computerware
P. O. Box 183
Grafton, MA 01519
Tel: (617)839-4134

GRAM Packer

Copyright ©1986 by J. Peter Hoddie
Version 1.1

Overview

GRAM Packer is a utility for use with GRAM Kracker and other GRAM devices which are becoming very popular these days. Essentially what it does is allow you to create your own custom menu of programs and cartridges that you use most often. The programs are accessed from the main menu and load in only a few seconds. There are also provisions for making programs available through the use of a CALL from BASIC or Extended BASIC, and also through regular file access from many cartridges and languages. You can decide where in GRAM the program will reside, what name should appear for it on the main menu, and a number of other parameters. Currently GRAM Packer can only pack a file saved for use with program image assembly loaders such as Editor/Assembler option 5, TI-Writer option 3, and the GRAM Kracker Load Cartridge option. Some programs use some strange tricks with these loaders which cause them to fail with GRAM Packer. However, all major programs tested including Funnel Writer, Fast-term, Disk Manager 1000, PR Base, TI-Artist, and Turbo Copy worked flawlessly.

Loading the Program

GRAM Packer can be loaded in a number of ways. For Editor/Assembler, select option 5 and enter DSK1.GP. For TI-Writer select option 3 and enter DSK1.GP. For GRAM Kracker select option 1, enter DSK1.GP and follow the on screen instructions. In all cases the program should start up fairly quickly.

Using the Program

After GRAM Packer loads, you will see a title screen. Press any key to go onto the program and you will be presented with the GRAM Packer main screen. First you will be asked for the first GRAM to use. You should enter a number from 1 to 7 depending on where you want to begin packing. Next you will be prompted for the last GRAM to use. This question is asked so that GRAM Packer can make sure that it doesn't fill in memory areas that are already in use. If you just press ENTER a 7 will be selected. Note that the last GRAM number must be greater than or equal to the first GRAM

number ... GRAM Packer only packs contiguous areas of memory.

After you tell GRAM Packer the area of GRAM that it can use, you will be prompted for the filename to pack. This can be any program image assembly file. Enter the filename just as you would to any other loader such as Editor/Assembler option 5. If the file is made up of multiple parts such as UTIL1, UTIL2, and UTIL3, you need only enter DSKn.UTIL1 as the rest will be loaded automatically.

Having entered the filename you will be prompted for a menu entry. This is the entry that will appear on the main menu. It can contain spaces but must not contain lowercase letters as these will not appear on the menu. You must enter a menu entry. If you just hit ENTER you will be returned immediately to this prompt.

When you have filled in all the required information (first and last GRAM, file name, and menu entry) you will be presented with a list of items near the bottom of the screen which let you set other optional parameters for GRAM Packer. To select one just press the first letter of the option. They are described below in alphabetical order.

Characters: This option toggles the Load Characters option. Some programs load their own character sets while others expect to use the character set that is already defined in the computer. Unfortunately, those programs that expect the character set to already exist will not find it because memory is arranged differently at the main menu than it is from Editor/Assembler option 5. Setting Load Characters on forces a complete set of upper and lowercase characters to be defined before the program you are packing is executed. If you choose to have characters loaded there are two disadvantages: (1) it takes more memory (14 bytes) to pack the code to load the characters and (2) your program will take about one second longer to start up. Some programs that require that characters be loaded (ON) are Disk Manager 1000, PR Base, and Funnelwriter. Two that don't have this requirement (OFF) are Fast-term and Turbo Copy.

DSR: This option stands for "Device Service Routine" and although it sounds fancy it is really quite simple. If you install a name here (such as FT for Fast-term) you can call up Fast-term by typing OLD FT from BASIC or Extended BASIC. From a running program you can call it up with DELETE "FT" or OPEN #5:"FT" or whatever. Any program that accesses peripheral devices through the console GROM DSRLNK routine (including both BASICs, most cartridges, Diskassembler, and many newer programs that use a fantastic DSRLNK written by Craig Miller and Doug Warren) will work this way. Any access to the peripheral through the name you define by DSR will cause the program to load into memory and begin execution.

Filename: This option just lets you re-enter the filename in case you made a mistake entering it the first time.

Menu: This option lets you re-enter the menu entry if you made a mistake the first time or change your mind.

Pack: This option starts the packing process. When you hit P, GRAM Packer goes to work setting up the cartridge header and then loading the appropriate file(s) into GRAM space. Before you select P make sure that the switches on your GRAM Kracker are all set to enable GRAM. This means that they should be: middle, up, up, up, down.

Quit: This option just takes you back to the master title screen. It is nothing more than a way out.

Subprogram: With this option you can define the name of a CALL routine to be associated with the file you are packing. For example, if you were packing PR Base you might call the subprogram PRB so that from immediate mode in TI BASIC or Extended BASIC, or from a running TI BASIC program, you can do a CALL PRB to start up PR Base. Note that if you define a CALL name that already exists such as FILES or HCHAR, more likely than not, the original call will be executed because of the way the operating system searches for CALLS.

TI Save: In 99.4% of the cases you want this turned ON.

All it says is that the file you are about to pack was created using the SAVE utility on the Editor/Assembler disk B. Unfortunately, this utility has a strange quirk that some people have corrected. The fix will not work with GRAM Packer unless you turn TI Save OFF. This is explained in more detail under "Advanced Applications." In general leave it turned ON, but if a file doesn't seem to pack correctly you might try turning this option OFF.

After you have filled in all the optional entries that you plan to use, hit P for Pack and GRAM Packer will do its job. If you are using a GRAM Kracker make sure that the switches are set as described under P above. If a file cannot be found or there is not enough memory available, GRAM Packer will issue an error message and ask you to "Hit SPACE to abort." If the pack is not successful you may end up with menu entries that appear but crash the system when they are selected. If this is the case you should re-initialize the GRAM space.

If the pack is successful, GRAM Packer will ask you if you want to "Pack another in this set?" If you are done, hit N and you will be instructed to restore write protection and will then be asked if you want to save the pack to disk. If you select Y, you will be prompted for a filename and then the pack will be saved to disk in GRAM Kracker format. If you select N, you will be returned to the master title screen.

If you have more to pack, and you believe there is enough memory free, you can select Y. This option allows you to pack several files into one block of GRAM, allowing for memory savings and greater flexibility. After selecting Y, all information about the last packed file will be cleared and you will be asked for a filename and menu entry before being presented with the option menu again. The ability to pack more than one file into a GRAM is particularly useful and powerful when used in conjunction with the utility options described below.

Saving Packed Files

If you chose not to save the packed files to disk after packing you can still save them using the built in GRAM Kracker software. To do this, just use the GRAM Kracker Loader software, SAVE option. This will save anything that you have packed in GRAMs 3, 4, 5, 6, or 7. If you have packed programs into GRAMs 1 and 2 (which are the most likely locations to pack into), you must use the Load/Save Console option to save GRAMs 1 and/or 2. Directions for doing this can be found in the GRAM Kracker manual. Because of the way that the GRAM Kracker save routine works, it is possible that it will not save all of your pack. If this is the case, you will have to repack and allow GRAM Packer to save the pack for you.

Memory Usage

What follows is general information on memory considerations when using GRAM Packer. A GRAM is 8K long. A general rule of thumb is that for each 33 sectors that are on the disk, one GRAM will be required. So don't try to pack 67 sectors into just one GRAM. It won't fit. However, it is possible that a 33 sector file will not fit in just one GRAM although this is unlikely. If you run into a problem with packing a file of 33 sectors (or series of files that are 33 sectors each) you might try shortening your menu entry, excluding DSR and Subprogram calls or shortening their names, as these things all take up memory. It is possible even after trying all these things, that the program will not pack into what appears to be just enough space.

One of the best locations to pack files is the GRAM 1-2 space. This space normally contains TI BASIC. Since most people choose to have Extended BASIC in their GRAM device TI BASIC isn't really needed anymore. The space left by TI BASIC is 16K long which is enough room to put Fast-term or Disk Manager 1000. Or enough room to include Funnelwriter in GRAM 2 and still leave the GRAM Kracker Loader ON. The other useful location for packing files is GRAM 7 which is not used by Extended BASIC or most other cartridges. This 8K segment could contain Funnelwriter, or any other 8K (or smaller) program, or several of the utility options below.

High Address Used

There is a piece of data on the GRAM Packer main screen that is titled 'High Adress Used.' This simply tells you what the highest GRAM address that GRAM Packer has used. It is sometimes helpful in determining if there will be enough room left to pack a file into. The number is displayed in hexadecimal and is updated after every GRAM write.

Testing a Pack

If you are not sure if there will be enough memory to pack a set of files into, you can perform a test pack before you actually pack by making sure that all banks of GRAM on the GRAM Kracker are write protected. To accomplish this set the five switches to: middle, down, down, middle, up. If you pack with the switches in these positions all the GRAM will be protected so nothing will be written but GRAM Packer will behave as if there was GRAM. Note that this technique will only work on a GRAM Kracker and not on other GRAM devices because of their lack of switches for write protection.

Trouble Shooting

Sometimes you will pack a file and find that it does not work. There could be many reasons for this. Some can be solved, others can not at the present time. Several possible problems with solutions are presented below:

- Program runs but characters look too big or no lower case. Try packing again with the Load Characters option ON.
- Disk drive light goes on and then nothing happens. Many programs that you can pack like the ARTIST1 file from TI-Artist are not complete programs, they are only a piece. When they start up they look for particular files on disk. Put the program disk of the file you packed in the drive and try selecting that option again from the menu.
- Program runs fine for a while and then dies. Program runs but not normally. Try re-packing with TI SAVE option OFF.

- Program starts up but then locks. Some programs use routines from TI BASIC. Try enabling TI BASIC and select the program again from main menu. Other programs try to load code directly out of the Editor/Assembler cartridge. Unless you load your Editor/Assembler cartridge into GRAM 3 (normal location) they will not execute correctly (if at all).
- Program stored entirely or in part in GRAMs 1 and/or 2 of the GRAM Kracker disappear or refuse to run correctly although when first packed they were fine. GRAMs 1 and 2 of the GRAM Kracker are not write protected and therefore often get wiped out when the system crashes or even sometimes at power up. The only solution to this is to reload or store the programs elsewhere.

There are many other possible problems, far too numerous to elaborate. If this seems discouraging, don't worry about it too much. In reality, most program will pack without any problems. However, there are enough exceptions that it is worth giving some general rules. If you have trouble packing a particular file, let us know. Maybe we can find a solution. Also, if you find a way to pack a tricky program, please let us know so that we can pass the information along to others who may face similar problems.

Utility Files

To allow for more flexible use of the GRAM Packer software, a special capability has been added to allow you to supply extra information to a packed file. If a file has been saved in a special way (described in the Advanced Applications section), it will prompt you for an "Additional filename" after packing is complete. This allows programs to be written so that a certain filename can be entered into it at a later time. Elsewhere in this manual there is information on creating this type of file, however in the following paragraphs several such files provided on the GRAM Packer disk are described followed by a specific example. The first one loads Editor/Assembler option 5 files, another loads GRAM Kracker cartridge and operating system files.

With the advent of RAM Disks, 80 track disk drives, and hard disks for the TI-99/4A many people have put together a utility disk containing their most frequently used utility programs. Furthermore, most people want to have Extended BASIC stored in their GRAM device. Unfortunately, Extended BASIC is a huge cartridge, requiring most of the available GRAM space. This severely limits the number and size of the programs that you can store in GRAM. An option has been provided to somewhat alleviate this situation. Using the file EA5;S;L (the filenaming convention used on the GRAM Packer disk is described under Advanced Applications), you can create a menu entry (along with DSR and subprogram entries) for a program that is stored on disk. The program itself will not be stored in GRAM but on your disk, RAM disk, or hard disk. When you select the file off the main menu it will be loaded into memory and executed. If you are using a regular disk drive, the only advantage in using this procedure is that you don't have to enter a filename. If you have a RAM disk the program loads nearly as quickly as if it were loading out of GRAM space. Thus you can create a large number of menu entries that take up relatively little memory (about 350 bytes each), that load very quickly, and still let you keep Extended BASIC (or whatever you like) in cartridge GRAM space.

Some people like to store copies of their most frequently used cartridges on their utility disk as well. This means that they must enable the GRAM Kracker loader software each time they want to load another cartridge, or must leave the GRAM Kracker loader switch ON all the time which means they aren't using GRAM 1 (where the GRAM Kracker loader software resides) for anything. Since the most common use of the GRAM Kracker software is to load in another cartridge and or GRAM 0, GRAM 1, or GRAM 2, it seems that there should be a less memory intensive way to load in a cartridge that would free up more GRAM space for other programs. On the GRAM Packer disk there is a file called GK;S that operates much like the EA5;S;L file described above does. Instead of loading an EA option 5 file from disk, it loads any file that was created using the GRAM Kracker internal software. Thus if you have a a number of frequently used cartridges stored on disk, you can create a number of menu entries to load them in. Because the loader contained in GK;S is written in assembly language instead of GPL it can be used to load any GRAM Kracker file including the operating system GRAM's without as much switch flipping as is required on the GRAM Kracker. It will prompt you to enable the correct banks of memory as the files are loaded.

Also on the disk are two general purpose loaders. One for E/A option 5 programs, one for GRAM Kracker files. They are called EA5;G;L and GK;G respectively. These files can be packed like any other files. You will not be prompted for a special filename. Instead when you select one of these programs from main menu you will be prompted for a filename. With the E/A option 5 loader, this file will be loaded in and executed. With the GRAM Kracker loader the file will be loaded, you will be prompted to enable the proper switches, and you will be asked for another filename. When you are done just hit ENTER and you will be returned to the main title screen. You can load any saved GRAM Kracker file using this option including GRAMs 0, 1, and 2.

There are also two files called EA5;G;H and EA5;S;H. These work the same as the similarly named files ending in ;L except that they load into high memory. In most cases you will want to use the ;L version but if that fails to load in a particular program you might try the ;H version.

If you wish to put Extended BASIC programs on the main menu, you may do so using the utility program XB. This program will save a short assembly program that makes Extended BASIC load a program other than DSK1.LOAD when it starts up. The approach used was devised by John Johnson in his excellent MENU program for the Horizon RAM disk. Please note that the XB file does not save the Extended BASIC program in memory, just a loader for it. Furthermore, Extended BASIC must be present for this to work (obviously!). This technique works 19 times out of 20. If it fails, just reset and try again.

As another method of saving some GRAM space, there is a file CALL on the disk, which is another Utility file. It allows you to simulate simple BASIC CALL's to peripherals. For example, the Horizon RAM disk has a CALL DM command. You can use this routine to install a CALL DM on main menu to load Disk Manager 1000. When prompted for an additional filename after packing this file enter the name of the call such as "DM". You cannot have more complex calls such as CALL RUN("DSK2.UTIL1") with this routine.

There are also two files on the disk for owners of the Maxi-Mem GRAM cartridge. These allow a GRAM Kracker saved cartridge to be loaded into Maxi-Mem. The files are called MM;G and MM;S and they behave just like the GK;G and GK;S files described above except that no switch flipping is required because of the way Maxi-Mem is set up. A similar utility for owners of the GRAM Karte will be made available if there is sufficient demand.

An Example

If you wanted to create an "indirect loader" using the utility option described above for the program Funnelwriter which is stored under the name UTIL1 on your RAM disk which is named DSK5, follow this procedure. Start up GRAM Packer, select the GRAM(s) you wish to pack to. When prompted for a filename enter DSK1.EA5;S;L (assuming that the GRAM Packer disk is in drive 1). When prompted for a menu entry, enter FUNNELWRITER. Now hit P to pack the file EA5;S;L. When this is done you will be prompted for an additional filename. For this enter DSK5.UTIL1. This will then be written to GRAM and you are done. When you select the option FUNNELWRITER from main menu, the program EA5;S;L will be moved out into the memory expansion and then EA5;S;L will load and run the file DSK5.UTIL1.

Multiple Menus

In the November 1986 issue of the Smart Programmer, Richard Mitchell hinted at an interesting method of creating multiple menus. Simply create a menu of 7 or 8 items in GRAM 2 and as the final option pack the file GK;S to load a file called MENU2 into GRAM 2. Save this pack as MENU1. Now start a new pack in GRAM 2, and as its final option pack GK;S to load MENU1 into GRAM 2. If you keep MENU1 and MENU2 on a RAM Disk they will load almost instantly and thus in effect you have two menus.

ADVANCED APPLICATIONS

TI Save Quirk

The SAVE utility provided on Editor/Assembler disk B claims to save program images in 8K (8192 byte) chunks. This is not quite true. It saves them in 33 sector files, which if you subtract out one sector as being dedicated to file header, gives 32 sectors of data storage (32 times 256 bytes per sector equals 8K). However, TI stores a 6 byte header before the actual program data which tells the loader how many bytes there are to load and where to put them. So actually TI stores 6 bytes less than 8K (8192 minus 6 gives 8186 bytes). In general this really doesn't matter, and actually helps to save disk space. However TI did one thing that doesn't quite make sense. In the 6 byte header the word that tells how many bytes of program data there are to load also includes the 6 header bytes in the count. This means that in effect this count is 6 bytes too long. Thus when GRAM Packer loads a file it subtracts 6 from the length so that it gets the correct number of bytes.

Some people (in particular the Funnelwriter folks) have written versions of TI's SAVE utility that do not count the 6 header bytes in the count of the length of the data. When GRAM Packer subtracts six from this number, it ends up missing six bytes of program data. "TI Save" option turns the subtraction of these six bytes on and off. There is no easy way to tell how the program was saved so this option was necessary. If the pack doesn't work one way, try it the other way. Fortunately, most programmers tend to use the official TI Save utility with its odd quirk.

Creating Utility Files

Creating a file to be used with the Utility option of GRAM Packer is a fairly straight forward procedure for the assembly programmer. What the utility option does is simply put a string preceded by a length byte directly after the packed file in memory. When the program is moved out to expansion RAM to execute this string is moved with it. So to create a program to use this option, you must create a program that can be loaded with E/A option 5 which means that it should include its own utilities such as VSBW, KSCAN, DSRLNK, XMLLNK, and the like. A source code file of these is included on the GRAM Packer disk for your convenience under the name UTILS/S. The program should recognize that in the bytes immediately

following the END directive there will be a string. Once you have

created a working program (obviously you will have to create a somewhat different method for obtaining the string for debugging purposes), save it with the TI Save utility. Then use a sector editor to change the first word of the first sector to >0377. This is the flag that tells GRAM Packer that this is a utility option type file so that it will prompt for a "special filename." Why >0377? Programmers are allowed their mysteries too. Source code for a simple example of a program to put a name on the screen using the utility option is on the GRAM Packer disk under the name NAME/S.

File Naming Conventions

The utility programs on the GRAM Packer disk have been named using a rather primitive but descriptive scheme. A ;S indicates that it is a "specific" to one file type of loader, which means it is used with the Utility option. A ;G indicates that it is a "general" program which will prompt for input. A ;L means that it loads into low memory and a ;H means that it loads into high memory.

Where a program loads in memory can be important in determining which version of a loader to use for a particular program. For example, if a program loads into low memory and the loader resides in low memory, then in the process of loading in the program the loader may be over written causing the system to crash. In such a situation it would be necessary to select a loader that resides in high memory.

GRAM Utilities

For those of you interested in writing your own assembly programs to access GRAM/GROM, there is a file on the GRAM Packer disk called GUTILS/S. It contains the assembly language source code to GMBW and GMBR routines which access GRAM/GROM just like VMBW and VMBR routines do for VDP access. It would be trivial to add routines for single byte access but there doesn't seem to be too much point in this since the majority of GRAM/GROM access is usually to read or write large blocks of data.

Some Suggestions on RAM Disks

If you are using a RAM disk it is very likely that you have it set up to operate as DSK1 so that you can use it with TI-Writer, Multiplan, auto load Extended BASIC programs, Editor/Assembler or whatever. Unfortunately this means that you can no longer use your real disk drive that is DSK1. There are a couple of solutions to this problem. First, if you access the disk by name instead of by number, the disk controller software will still find the file on the physical disk. For example if the disk in DSK1 is named FRANK and you want to load the file GAME, instead of typing OLD DSK1.GAME, enter OLD DSK.FRANK.GAME. With a CorComp disk controller you can access the disk by the letters DSK1 in lower case such as dsk1, however you must still enter the filename in uppercase.

If you are using a Myarc RAM Disk with a GRAM Kracker you have probably noticed that the RAM Disk contents get corrupted rather often. This problem is a result of reset switch on the GRAM Kracker which apparently sends some sort of spike through the entire system. So try to avoid using the reset switch and your RAM disk may survive a little longer.

Random Notes

GRAM Packer writes GRAM through address >9800. If you are using a GRAM Karte you must make sure that it is set to respond at address >9800 or GRAM Packer won't write anything to the GRAM space. The GP file which is the GRAM Packer program was created using the TI SAVE utility so that if for some reason you choose to pack it, the TI SAVE option should be ON. However, it does not load in its own characters so that if you want to pack it you must make sure to set Load Characters ON. GRAM Packer was inspired by Barry Traver's request for someone to find a way to install Fast-term or Disk Manager 1000 in a GRAM Kracker. Having done that twice "by hand" I decided there had to be an easier way. That's supposed to be this program. Many thanks to Walt Howe and Barry Traver for testing out this program and making many helpful suggestions. And particular thanks to Doug Warren and Craig Miller for creating a piece of hardware as wonderful as the GRAM Kracker.