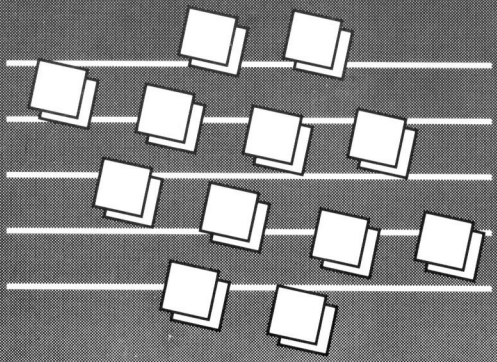
U·S·E·R'·S G·U·I·D·E

FAAASTROM™

Scanned by Josh



GVP

GREAT VALLEY PRODUCTS INC.

This manual is the Copyright © 1990 of Great Valley Products, Inc. All Rights Reserved. This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form, without prior consent, in writing, of Great Valley Products, Inc.

GVP, Impact, Impact Series II, DPRC, and FaaastPrep are registered trademarks of Great Valley Products, Inc.

Amiga is a registered trademark of Commodore-Amiga, Inc. Commodore is a registered trademark of Commodore Electronics Limited. Macintosh is a registered trademark of Apple Computers, Inc.

TABLE OF CONTENTS

NOTE:

Please remember to send in your warranty registration card.

Partitioning and Formatting Using FaaastPrep

Included with your FaaastROM Kit is GVP's new EPROM based driver. This new driver is contained in only one EPROM and takes the place of the EPROM(s) you are now currently using.

Warning: *When installing* EPROMs *it is essential that they are installed with proper orientation. There is a notch on one end of the* EPROM *which must go in the same direction as the white labeling on the board itself. If an* EPROM *is installed backwards it will destroy the* EPROM. *If you feel uncomfortable doing this yourself then have your dealer install it for you.*

If the driver you purchased is SCSI and your old driver is contained in *two* EPROMs then you must replace the old EPROM labeled EVEN with this new EPROM and remove and discard the old EPROM labeled ODD. This means that the new EPROM will be in the socket once occupied by the old EVEN EPROM and the old ODD EPROM socket is now empty.

If the driver you purchased is SCSI and your old SCSI driver is in *one* EPROM then simply replace the old EPROM with this new EPROM.

If the driver you purchased is for the AT interface on our accelerator boards then discard the EPROM(s) currently in locations U34 and U33 (if applicable) and insert the new EPROM in location U34.

In order to utilize the full power of FaaastPrep, we will first give you a brief overview of this utility. When executing this utility by double clicking on the FaaastPrep icon, you will be presented with a startup screen offering you several options. Your first choice is whether you are going to use this utility on your SCSI drive or your AT drive. In order to switch between these two options, there is a small gadget in the middle of the screen.

The next option concerns automatic versus manual installation. The following is a brief description of both choices:

Automatic: This is the easiest of the two selections when installing a hard drive. This option requires a minimum of input from the user and is the fastest way to set up your hard drive. The Automatic Installation is appropriate for novice users, as well as for those who desire a rapid hard drive installation without the immediate need for special options.

Manual: This option gives the experienced user total control over their hard drive. Parameters such as boot priority, max transfer, and buffer size may be altered for any partition even after the drive has been previously partitioned and already contains data. Other parameters such as low and high cylinders are also alterable. The manual mode is recommended for the experienced user only.

Note: *Hard drives previously prepped using the old* GVP *drivers can be re-prepped and updated for the new driver using the* Automatic *mode. Using the* Automatic *mode to re-prep your hard drive will retain any data currently on the drive.*

Automatic Mode Instructions:

Once you have chosen SCSI or AT you must then click on the Automatic gadget. You will then be presented with the installation screen for the automatic mode. FaaastPrep will default to drive ID number zero, if you are installing a drive other than ID number zero, click on the "Skip to Next" gadget to proceed to the next available unit.

Once you have chosen your correct ID number, FaaastPrep will read in some information regarding the drive and display it on the left hand side of the screen. Now, enter the number of partitions into which your drive will be divided by clicking on the appropriate gadget and entering your desired value. Followed by a return. Please note that due to limitations in the Amiga's file system, only one partition is allowed on removable media devices. (Pleasee refer to the chapter titled "Programmer's Documentation" for more information.)

Next, FaaastPrep will automatically divide the total storage capacity into equal sizes. If you do *not* want the partitions to be the same size, all you have to do is to position the cursor in the partition size box and alter the number within the box. Always finish with the RETURN key. Please note that in the lower left hand corner of the screen are two drive size indicators for your reference.

Once you have entered the number of partitions and their respective sizes, you must decide the partition names. FaaastPrep will offer you default partition names which you may change or accept. If you change the names, just position the cursor in the partition name box and re-type the name followed by a return.

Note: *Prep/Format will erase all data currently on the target drive.*

Once everything is to your preference, click on the "Prep/Format" gadget to send this structural data to your drive. Once FaaastPrep is finished, the drive will be prepped and formatted.

Software Installation:

Once your hard drive has been installed and initialized, it will be ready to accept software. Your hard drive partitions act just like large floppy disks, only much faster. This means that copying to and from the hard disk is the same as if you were copying between floppy diskettes.

Some software requires special hard drive installation routines to access them from a hard drive. It is recommended that you consult the software manual that came with the software on how to install it on a hard drive.

You will also want to consult your AmigaDOS manual on additional information regarding copying.

Manual Mode Features

Low Level Format: This will instruct a hard dirve to complete a low level format on itself. Low level format will also present a screen allowing you to enter the interleave for the target drive. An interleave of zero(0) tells the drive to use its own default setting for interleave. Any other number will create an interleave of that number to one (1) (i.e. 1=1:1, 2=2:1, 3=3:1, etc.). Note that certain drives such as Quantum and Maxtor do not allow you to change the interleave value, they always default to 1:1. Also, low level format will completely erase the drive.

Note: *This function is not implemented for* AT *drives.*

Bad Block Re-Map: This will re-map any defective blocks that a drive may have while keeping the data intact. This re-map will attempt to restore any data found on the bad blocks themselves. Also, you should always back up your hard drive prior to a bad block re-map.

Note: *This function is not implemented for* AT *drives.*

DOS Format: This is the equivalent of the AmigaDOS format.

Read: Reads the rigid disk block and the partition block of the target hard drive.

Write: Writes the data you specified to the rigid disk block and the partition block of the target drive.

Unprep: Erases the first 64 blocks, removing the rigid disk boot block and the partition blocks.

Auto: When enabled, this will automatically adjust cylinders, heads, and sectors according to your specification. (i.e. When entering sectors FaaastPrep will automatically re-calculate high cylinders.) This will also automatically calculate high and low cylinder values according to partition size values.

FileSys: File system type. FFS-Fast File System, OFS-Old File System, AFS-Alternate File System (currently not utilized).

Mem: Type of memory that the buffers are allocated from. E-either fast or chip, C-Chip, F-Fast.

Mask: Address Mask to specify memory range that DMA transfers can use.

< >: Use these gadgets to change the target drives ID number.

USING THE PC BRIDGEBOARD WITH A GVP HARD DISK

The GVP Impact Series II controller boards support Commodore's A2088 and A2286 Bridgeboards by providing a convenient way to create ***virtual drives*** for the PC side to access. The term ***virtual drive*** refers to the process by which an AmigaDOS file is created on the Amiga side to act as a separate hard drive for the PC side. In this way, a GVP-controlled SCSI hard drive offers you fast, effective storage for both your Amiga and your PC applications.

There are presently two versions of the Janus Software, the essential installation software accompanying your Bridgeboard. In this section of our manual, GVP has attempted to clear up the confusion that formerly surrounded this process. **Please check your Janus software to determine whether it is Version 1 or Version 2**, and follow the corresponding procedure for the rapid creation of your virtual drive.

NOTE: The following assumes that you have:

- successfully completed the hardware installation of the Bridgeboard by following the instructions in its manual.
- successfully completed the hardware installation of your GVP hardcard OR controller card/cable/drive assembly — please refer to the appropriate chapter in the table of contents.
- successfully set aside a formatted AmigaDOS partition of your hard drive that will contain the PC virtual drive file. Follow the instructions for software installation in Chapter 9. We recommend that the size of this allocated partition be between 5 and 20 megabytes, depending on how much of your hard drive space you are willing to allocate to your PC side. (Please note also that the MS-DOS limit is 32 megabytes).

INSTRUCTIONS FOR JANUS VERSION 1

Step 1:

Insert the GVP Installation diskette in your floppy drive and double click on its icon. Next, run GVP's Bridgeboard Install utility by double clicking on the BBI icon. Follow the instructions prompted by the BBI utility. When BBI is finished, proceed to the next step.

Step 2:

The next step involves adding a single line to the "startup-sequence" file in the "s" directory so that the virtual drive information is fed to the Amiga side upon boot-up. This command line may already be present in your "startup-sequence", so you must check carefully as we proceed. Type

ed sys:s/startup-sequence

to enter into the startup-sequence file. Cursor down through the various lines to find a line that reads, **Binddrivers**. If **Binddrivers** is *not* present, you should add it about 7-10 lines from the top (from the bottom if you have an accelerator card installed) by typing a single RETURN at the end of the line you wish **Binddrivers** to follow, and then typing **Binddrivers** into the newly-created empty line. (Refer to the AmigaDOS manual for instructions regarding ED.)

Next, place the cursor after the **Binddrivers** command and press the return key to insert a new line after it. Now type into the empty line the following:

run >nil: sys:pc/pcdisk.

This routine allows the Bridgeboard to read and write Amiga drive information. To exit, again hit **ESC**, and then **X**, and then RETURN to save the modified startup-sequence. (Note: If you enter a **Q** instead of an **X** all your edits will be lost and you will be able to start again.)

Step 3:

Place the MS-DOS System Disk#1 in the 5.25" IBM drive. Reboot your computer (CTRL-AMIGA-AMIGA). Open the PC color or PC mono window — depending on what jumper settings you have selected by consulting your Bridgeboard manual.

Step 4:

Always make backups! You can make a backup of the 5.25" MS-DOS disk by typing "**diskcopy a: a:**" at the "A>" prompt. **NOTE: The quotation marks used throughout this section are only for your reference; they are not to be typed.**

Step 5:

Store the original MS-DOS diskette in a safe place, and make sure its COPY is in the 5.25" drive. This is your MS-DOS boot disk, and you will now modify it to facilitate the use of a virtual drive. Still within the PC window, type the following at the "A>" prompt being careful to use the correct backslash key.

Copy con a:\config.sys

On the screen, you will receive a new line without a prompt, at which point you should type the following lines of text exactly as they are shown below. The comment column separated with semicolons is just for your reference and is NOT to be entered. After each separate line, press the RETURN key.

YOU TYPE	;	COMMENT
files = 20	;	Maximum number of open files
buffers = 20	;	Reserved memory for drive speed
device = jdisk.sys	;	Loads PC-Amiga translator
device = ansi.sys	;	Loads common graphics driver
^Z	;	End of this file. * See next page *

* The "**^Z**" or so-called "**control Z**" character is obtained by holding the control (CTRL) key down and pressing the Z key at the same time. After this last line is typed and the RETURN key is pressed, the computer will give you a confirmation message, and your prompt should return to the screen.

***Step* 6:**

Reboot the Bridgeboard by pressing the keys CTRL, ALT, and DEL simultaneously. Once the "A>" prompt has returned, you are ready to create the virtual drive. Before you enter the appropriate command, you must first determine four things:

1. **MS-DOS virtual drive letter**— (C,D,E, or F) by choice
2. **AmigaDOS drive number**— partition set aside (i.e.dh1,dh2...)
3. **AmigaDOS filename**— by choice (i.e. pcdrive, mydisk, etc.)
4. **Virtual drive size**— in kilobytes (i.e. 15meg=15000)

Our example drive created below uses the first example selection from each of the four given options above. You will now enter the following line while substituting your own names and numbers in place of the example selections in bold:

jlink **c: dh1:pcdrive /c:15000**

Be certain that you have pressed RETURN after this line, and **be aware that if you are recreating an already established virtual drive that all of the MS-DOS data will be erased.**

***Step* 7:**

Transfer the contents of the 5.25" floppy diskette to the newly-created virtual drive by entering:

copy a:*.* c:

***Step* 8:**

Now you need to make a boot file on the same floppy that will serve to automatically activate the virtual drive and transfer control to it after each startup. Please type

Copy con a:\autoexec.bat

and hit RETURN. Next, enter the following lines without typing the semi-coloned comments which are again provided only for your information.

YOU TYPE	COMMENT
echo off	; limits output to the screen
cls	; clears the screen
jlink c: dh1:pcdrive	; re-links virtual drive *see below*
set comspec=c:\command.com	; DOS to virtual drive
path=c:	; sets path to virtual
c:	; defaults to the c drive
^Z	; ends the file, like

***Step* 9:**

You have now completed your virtual drive installation. * In the above example, MS-DOS drive "c" is the virtual hard drive contained on a file called **dh1:pcdrive** of the Amiga side; you should use the appropriate drive names and filename when you enter this line. **NOTE: There are certain precautions that you should always follow to insure the future proper operation of your virtual hard drive:**

1. **Always UNLINK** the drive before rebooting or turning off your computer by typing **"jlink c: /u"** where "c" is the drive letter.

2. **As an added security,** before you use the virtual drive for your applications, fill it completely with any data and/or files, unlink it, then re-link it and delete this data. This procedure expands the virtual drive before you place valuable data on it, and an accidental reset is much less likely to damage your information later on.

INSTALLING THE JANUS SOFTWARE VERSION II

***Step* 1:**

Insert the GVP Installation diskette in your floppy drive and double click on its icon. Next, run GVP's Bridgeboard Install utility by double clicking on the BBI icon. Follow the instructions prompted by the BBI utility. When BBI is finished, proceed to the next step.

***Step* 2:**

The next step involves adding a single line to the "startup-sequence" file in the "s" directory so that the virtual drive information is fed to the Amiga side upon boot-up. This command line may already be present in your "startup-sequence", so you must check carefully as we proceed. Type

ed sys:s/startup-sequence

to enter into the startup-sequence file. Cursor down through the various lines to find a line that reads, **Binddrivers**. If **Binddrivers** is *not* present, you should add it about 7-10 lines from the top (from the bottom if you have an accelerator card installed) by typing a single RETURN at the end of the line you wish **Binddrivers** to follow, and then typing **Binddrivers** into the newly-created empty line. (Refer to the AmigaDOS manual for instructions regarding ED.)

Next, place the cursor after the **Binddrivers** command and press the return key to insert a new line after it. Now type into the empty line the following:

run >nil: sys:pc/pcdisk.

This routine allows the Bridgeboard to read and write Amiga drive information. To exit, again hit **ESC**, and then **X**, and then RETURN to save the modified startup-sequence. (Note: If you enter a **Q** instead of an **X** all your edits will be lost and you will be able to start again.)

***Step* 3:**

We will now format the virtual drive and copy the contents of the MS-DOS floppy disks over to it. Insert the MS-DOS system disk number 1 into the 5.25" drive and reboot the Amiga (CTRL-AMIGA-AMIGA). Open the appropriate color or mono window from the PC drawer. At the "**A>**" prompt, type "**FDISK**" and hit RETURN. You will be presented with an option menu. Select option #1 to create a DOS drive partition. Next choose option #1 again to create a *primary* DOS partition. Next, designate an ***active*** partition as the number 1 partition. When this has finished, hit the escape key to exit, and then reboot the Bridgeboard side with (CTRL-ALT-DEL). Some versions of the A2088 board might "hang" the system, in which case you should use the "PCHard" icon to reset it.

***Step* 4:**

At the new "**A>**" prompt, you will format the virtual drive by typing, "**format c: /s**", being careful to use the proper slash key. When that finishes, type "**xcopy a: c: /s**" to copy the first system disk to the virtual drive. When the copy is complete, type "**C:**" to make the virtual drive the current directory. Place the second MS-DOS disk into the 5.25" floppy drive, and retype "**xcopy a: c: /s**" to copy the contents of the second floppy to the virtual drive. Repeat this process once more with the third MS-DOS system disk. You have now completed the installation. You may test your work by removing the 5.25" floppy and rebooting the Bridgeboard side. Now, watch your GVP-controlled hard drive autoboot as a virtual MS-DOS drive!

PROGRAMMER'S DOCUMENTATION

In this chapter, we will explain some of the more technical details related to GVP's SCSI software which are useful to know if you want to program the SCSI driver or if you want to improve the performance even further. Some sections (and the example programs on the distribution disk) assume some familiarity with the Amiga's I/O system, especially the way the trackdisk.device works. For an introduction, see [3, Chapter 19 and Chapter 42].

Initialization

The GVP device driver must be opened via the exec.library's OpenDevice() call. It requires a standard I/O request, the "flags" parameter must currently be set to zero. Here is an example:

```
struct MsgPort *mp;
struct IOStdReq *io;

If((mp = CreatePort(NULL, 0)) != NULL)
 {
 If((io = CreateStdIO(mp)) != NULL)
  {
  If(OpenDevice("gvpscsi.device", UNIT, io, 0) == 0)
   {
   /* SCSI commands go HERE */

   CloseDevice(io);
   }
  DeleteStdIO(io);
  }
 DeletePort(mp);
 }
```

GVP's SCSI driver adheres to the standard defined in the include file <devices/scsidisk.h>, so the unit number has the following meaning: The 100s digit is the board number (our driver supports any number of boards), the 10s digit is the logical unit number (LUN), and the 1s digit is the SCSI target ID. The host adapter itself currently uses target ID 7 (HFERR_SelfUnit on OpenDevice()), but this is subject to change. For further information, see [2, pages D-16 and D-17] and [2, pages E-17 and E-18]. The driver automatically recognizes all GVP SCSI controller boards in the system.

Trackdisk Commands

GVP's SCSI driver supports the following trackdisk.device commands (see [2, pages B-70 to B-76] for the official documentation on the trackdisk.device and [2, pages D-19 and D-20] and [2, pages E-20 and E-21] for the corresponding include file definitions):

CMD_READ
CMD_WRITE
TD_FORMAT
TD_MOTOR
TD_SEEK
CMD_UPDATE
CMD_CLEAR
TD_REMOVE
TD_CHANGENUM
TD_CHANGESTATE
TD_PROTSTATUS
TD_ADDCHANGEINT
TD_REMCHANGEINT

gvpscsi.device will automatically support *any* block size up to 64 kilobytes that is a power of two, so the io_Length and io_Offset parameters for the CMD_READ, CMD_WRITE and TD_FORMAT commands do not necessarily have to be multiples of TD_SECTOR (512) bytes, even though the 1.3 file systems (ROM and FFS) do not support other block sizes. It is possible, however, to use SCSI devices with a block size *smaller* than 512 bytes (e.g. 20-meg Iomega Bernoulli with 256-bytes sectors), since our SCSI driver will handle requests of any size as long as it is a multiple of the current logical block size as understood by the SCSI drive. This is also the reason our new driver does not require the MountList MaxTransfer kludge as most other drivers do (i.e. MaxTransfer is 0x7FFFFFFF). This results in much faster transfer of large data blocks.

In contrast to trackdisk.device V1.3 or earlier, TD_REMCHANGEINT *does work*! Use of TD_REMOVE is not recommended, since this might interfere with the file system. The current versions of the Amiga file systems do not make use of the TD_ADD/REMCHANGEINT calls, since many drivers do not support them or simply crash. Therefore you should not use more than one partition on any removable medium!

Always make sure the motor is turned off using TD_MOTOR after CMD_READ, CMD_WRITE or TD_FORMAT, since otherwise you might not be able to eject removable media cartridges (e.g. SyQuest).

Direct SCSI *Commands*

The Commodore SCSI standard provides for a way of sending raw SCSI commands that do not fit into the Amiga's I/O system to the device. This way, a programmer can use any command supported by the SCSI device's firmware. Here's an example on how this feature can be used:

```
struct IOStdReq *io;
struct SCSICmd SC;
UBYTE command[12];

io->io_Command      = HD_SCSICMD;
io->io_Length       = sizeof(struct SCSICmd);
io->io_Data         = (APTR)&SC;

SC.scsi_Data        = (UWORD *)scsidata;
SC.scsi_Length      = scsilength;
SC.scsi_Command     = command;
SC.scsi_CmdLength   = 6;
SC.scsi_Flags       = SCSIF_READ | SCSIF_AUTOSENSE;
SC.scsi_SenseData   = sensedata;
SC.scsi_SenseLength = MAXSENSE;
SC.scsi_SenseActual = 0;

command[0] = scsicmd;
command[1] = LUN << 5;
command[2] = 0;
command[3] = 0;
command[4] = scsilength;
command[5] = 0;

DoIO(io);

/* ... */
```

HD_SCSICMD is the name of the Exec command being used, io_Length is always set to sizeof(struct SCSICmd), io_Data points to the SCSICmd structure. The SCSICmd structure in turn points to the memory area for the SCSI data and the Request Sense data. It also contains fields for data length and SCSI status information.

For a complete example on HD_SCSICMD, see the distribution disk. Make sure you also read and understand the section "Memory Requirements". One more restriction: Don't use HD_SCSICMD to perform block I/O on disk devices, use the trackdisk commands (CMD_READ, CMD_WRITE) instead!

The SCSIF_AUTOSENSE feature requires the use of the latest version of <devices/scsidisk.h>, which is, unfortunately, not included with the Version 5.05 of the Lattice C compiler. It can, however, be obtained from Commodore Applications and Technical Support (CATS) as part of the Native Developer Update Kit 1.3.

Memory Requirements

There are certain restrictions on the memory regions that can be used when dealing with SCSI DMA controllers. These restrictions do not necessarily apply to the GVP SCSI controller boards and/or drivers, they should, however, be taken into account when writing software that is supposed to work with other manufacturer's SCSI controllers.

Memory regions used for DMA should be at least word-aligned, some controllers (none of GVP's though) even require longword-alignment. This is easily achieved with the AllocMem() call. The other requirement is related to the type of memory used: Since most SCSI controllers are Zorro-II devices (i.e. designed for the A2000 with its 24-bit address bus), they can only DMA into the lower 16 megabytes of the Amiga's address space. If you happen to have a RAM expansion board that is located outside the address range $00000000 to $00FFFFFF or one that does not support DMA, you must restrict all SCSI data transfers to those regions of memory that do support DMA. In the worst case, this is Chip memory only, so you should allocate data buffers with the MEMF_CHIP attribute and set the MountList Mask-parameter to 0x1FFFFF (Chip memory only). Unfortunately, there is no way a programmer can request Zorro-II memory

without traversing the free-memory list. None of these restrictions apply to the GVP Impact Series I SCSI controllers, and usually it is sufficient to use a Mask value of 0xFFFFFE. Please refer to your RAM expansion manual for further information.

Summary: For CMD_READ, CMD_WRITE and TD_FORMAT, io_Data should be at least word-aligned and point to DMA-able memory. io_Length should be an even number of bytes. For HD_SCSICMD the same restrictions apply to scsi_Data, scsi_SenseData, scsi_Length and scsi_SenseLength. scsi_Length should not exceed the 24-bit limit.

MountList and the Rigid Disk Block

GVP's SCSI software adheres completely to Commodore's Rigid Disk Block standard. This allows automounting/autobooting with alternate file systems (non-ROM file systems). We also support the FileSystem.resource and file system version numbers, so the SCSI driver will automatically use a later ROM file system if one is available. This enables you to use the Kickstart 2.0 ROM file system and all of its new features, even though the drive might have been prepped with the 1.3 FFS. It also requires much less memory.

One other advantage of the RDB standard is that you can move hard drives and cartridges between different manufacterers' controllers and boot from them without having to re-prep anything.

VUP*rep*

The RDB standard is defined in [2, pages D-5 and D- 6], [2, pages E-6 and E-7], and [3, pages 538-545].

GVP SCSI *the* MMU *and* V. 2.0

Version 3 of our SCSI device driver no longer requires a SetCPU CARDROM entry, since the driver will copy itself to 32-bit memory (if any is available) and therefore execute at the highest possible speed. Since SetCPU requires large memory blocks, this will save valuable 32-bit memory and leave the MMU available for other tricks.

Speeding up the Autoboot Process

During autoboot, the SCSI device driver scans the SCSI bus for all devices currently connected. Unfortunately, this takes a few seconds, so you might want to break out of this loop if you do not have the maximum of seven SCSI devices connected to your Amiga. This can be done by setting the "Last Disk", "Last LUN" or "Last TID" flags of the RDB, which will result in a slightly faster autoboot process.

Removable Media

gvpscsi.device fully supports removable media (e.g. SyQuest, Iomega Bernoulli, Ricoh) by implementing the trackdisk commands TD_REMOVE, TD_ADDCHANGEINT and TD_REMCHANGEINT. The 1.3.2 Fast File System, however, does not make use of them, so you would really have to use the old ROM file system if you wanted to use removable media. Since this is not an acceptable solution, our SCSI driver uses some special tricks to tell the FFS about media changes. One drawback, however, still applies: Workbench icons might change to "BAD" instead of disappearing after you remove a cartridge.

GvpMount is a small utility that can be used to mount removable media that have not been inserted at boot-time. Simply call it without any command line options. Option "-s" will cause GvpMount to stick around until all SyQuests etc. have been mounted (default if you start it from Workbench). This way it can be "Run" in your "Startup-Sequence".

Disconnect

Version 3 of the GVP SCSI device driver automatically enables all SCSI devices to "disconnect" if more than one is connected. This means that if a SCSI device detects that an operation might take longer than usual (due to seeks etc.), it will release the SCSI bus and give other SCSI devices a chance to operate. Imagine an Amiga with a hard drive and a tape unit: Without the disconnect feature, every time your backup-software would rewind the tape, you could do abolutely nothing with your hard drive until the operation was complete! The same applies for example to low-level formatting of hard drives, which usually takes a few minutes. Under gvpscsi.device, the SCSI will send the driver a message that it is releasing the bus and yet another message when the low-level format is done. Between these two messages, all other SCSI devices can operate normally, they could even disconnect as well. This is similar to the Amiga's multi-

tasking ability where one task releases the processor (i.e. "goes to sleep") until an external event takes place, thereby giving other tasks a chance to run.

The disconnect feature has a slightly higher per-command overhead than the "nasty" mode which "holds" the SCSI bus until the command is completed. With the GVP Impact Series I SCSI controllers, writing very large blocks of data can also be slightly slower in disconnect-mode. If you do not normally operate with several SCSI devices at the same time, you might wish to disable the Disconnect/Reselect feature. If you only have a single SCSI device connected, the driver will use the nasty mode by default.

References:

[1] American National Standard for Information Systems - Small Computer System Interface (SCSI), ANSI X3.131-1986, American National Standards Institute, Inc.

[2] Commodore-Amiga, Inc.: Amiga ROM Kernel Reference Manual: Includes & Autodocs, 1989, ISBN 0-201-18177-0, Addison-Wesley Publishing Company, Inc.

[3] Commodore-Amiga, Inc.: Amiga ROM Kernel Reference Manual: Libraries & Devices, 1989, ISBN 0-201-18187-8, Addison-Wesley Publishing Company, Inc.

WARRANTY

For a period of one year from the date of purchase, Great Valley Products warrants to the original purchaser that the hardware shall be free from manufacturing defects.

This warranty is applicable only to the original purchaser who purchased the product directly from Great Valley Products or from an authorized dealer. This warranty shall be void if the unit has been subjected to improper or abnormal use, or if modified, or dismantled in any way. If a defect occurs during the warranty period the unit must be returned, postage paid, in original packaging, to the place of purchase, together with the sales receipt.

The Purchaser's sole and exclusive remedy in the event of defect is limited to the correction of the defect by adjustment, repair, replacement, or refund at Great Valley Products' election. Great Valley Products will in no event be liable for any direct, indirect, or consequential damages to any equipment used with any hardware or software sold by Great Valley Products.

Any software provided is sold or licenced "AS IS". The entire risk as to its quality, performance, or fitness for any particular purpose is assumed by the buyer. Should the software prove to be defective following purchase, the buyer assumes the entire cost of all necessary servicing, repair, or correction of any incidental or consequential damages.

Any statements made by Great Valley Products and its employees, including but not limited to, statement regarding capacity, suitability, or performance of the unit shall not be deemed a warranty or representation of Great Valley Products for any purpose, nor give rise to any liability or obligation of Great Valley Products.

GREAT VALLEY PRODUCTS, INC.

600 Clark Avenue • King of Prussia, PA 19406 • USA
TEL: (215) 337-8770 • FAX (215) 337-9922