Intelligent Computer Peripherals ®

ICP

Chapter C

Quick-Setup

C. Quick-Setup

C.1 What is the Aim of Quick-Setup ?

In the previous chapter we installed the ICP Controller in a PCI computer and connected the SCSI devices. Now these SCSI devices must be prepared in order to run with your operating system. This **Quick-Setup** chapter should help you to get started quickly. Quick-Setup shows **four examples** on how a single SCSI hard disk, a Mirroring Array Drive (RAID 1), a RAID 5 Array Drive and a RAID 5 Array Drive with a Hot Fix drive are installed:

Example 1: Installing a single SCSI hard disk.

Example 2: Installing a Mirroring Array Drive (RAID 1), consisting of two SCSI hard disks.

Example 3: Installing a RAID 5 Array Drive, consisting of five identical SCSI hard disks.

Example 4: Installing a RAID 5 Array Drive, consisting of four identical SCSI hard disks, and adding one Hot Fix SCSI hard disk.

Examples 3 and 4 are not applicable to ICP Controllers without the *RAIDYNE* firmware. Even if you cannot practically carry out all the examples yourself, we suggest reading them all the same because they will give you a better understanding of how the controllers of the GDT RP Series work. The following table tells you which examples are applicable to your type of ICP Controller.

	Example 1	Example 2	Example 3	Example 4
GDT6111RP	Yes	Yes	No(*)	No(*)
GDT6121RP	Yes	Yes	No(*)	No(*)
GDT6117RP	Yes	Yes	No(*)	No(*)
GDT6127RP	Yes	Yes	No(*)	No(*)
GDT6511RP	Yes	Yes	Yes	Yes
GDT6521RP	Yes	Yes	Yes	Yes
GDT6517RP	Yes	Yes	Yes	Yes
GDT6527RP	Yes	Yes	Yes	Yes
GDT6537RP	Yes	Yes	Yes	Yes
GDT6557RP	Yes	Yes	Yes	Yes

(*) **Yes**, when *RAIDYNE Upgrade* installed.

With examples 3 and 4 we shall briefly repeat the installation of the ICP Controller and the SCSI devices, in particular with regard to disk arrays.
Some essential issues having direct impact on the structure and configuration of an Array Drive with RAIDYNE will also be discussed:

1. How many physical SCSI hard disks are to be integrated in the Array Drive ?
2. Which redundancy level ought to be achieved ?
3. Should RAIDYNE automatically recover redundancy in the event of a disk failure ? Or, in other terms: Are Hot Fix drives needed ?

Before we go through these examples step by step, we would like to explain a few terms and relations important for the basic understanding of the ICP Controller firmware. At the end of example 4, we will try to answer the three questions above.

C.2 What is the ICP Controller Fimware ?

We refer to firmware as the operating system which controls the ICP Controller with all its functions and capabilities. The firmware exclusively runs on the ICP Controller and is stored in the Flash-RAM on the ICP Controller PCB. The controlling function is entirely independent of the PCI computer and the host operating system installed (for example UNIX), and does not "drain" any computing power or time from the PCI computer. According to the performance requirements needed, the ICP Controllers are available with two firmware variants. The firmware is either already installed on the controller upon delivery, or can be added as an upgrade: *RAIDYNE upgrade.*

- **Standard Firmware** (installed on the GDT6**1**xyRP controllers).
  In addition to simple controlling functions regarding SCSI hard disks or removable hard disks, this version allows disk chaining (several drives can be linked in order to form a single "large" drive), and the configuration of Array Drives of the types data striping (RAID 0) and disk mirroring or duplexing (RAID 1).
- **RAIDYNE Firmware** (installed on the GDT6**5**xyRP controllers). In addition to disk chaining, RAID 0 and RAID 1, RAIDYNE allows you to install and control Array Drives of the types RAID 4 (data striping with dedicated parity drive), RAID 5 (data striping with distributed parity) and RAID10 (a combination between RAID 0 and 1)

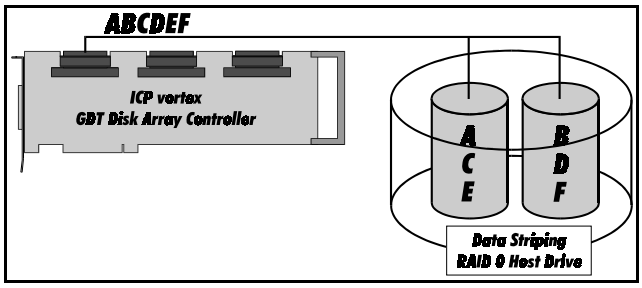
RAIDYNE is the name of the ICP disk-array operating system for the ICP Controllers. Unlike pure software solutions, RAIDYNE is totally independent of the host operating system, and can therefore be accessed under MS-DOS, Windows, OS/2, SCO-UNIX, Interactive UNIX, Novell NetWare, etc.. Special RAID drivers are not needed. The integration of a RAID Disk Array into the host operating system is carried out with the same drivers used for the integration of a single SCSI hard disk. All ICP Controllers are equipped with a hardware which is particularly well suited for disk arrays. RAIDYNE uses this hardware with extreme efficiency and therefore allows you to configure disk arrays that do not load the host computer (whereas all software-based RAID solutions more or less reduce the overall performance of the host computer.).

The basic concept of the RAIDYNE is strictly modular, and consequently, in its functioning it appears to the user as a unit construction system.

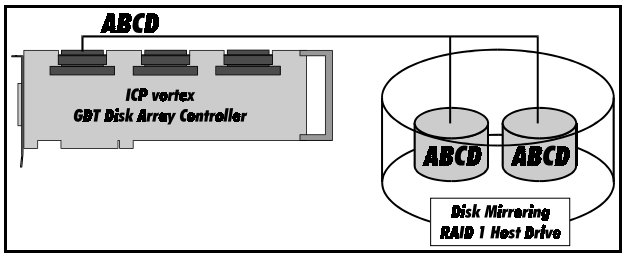
C.2.1 The Different RAID Levels

RAID 0 - Data Striping

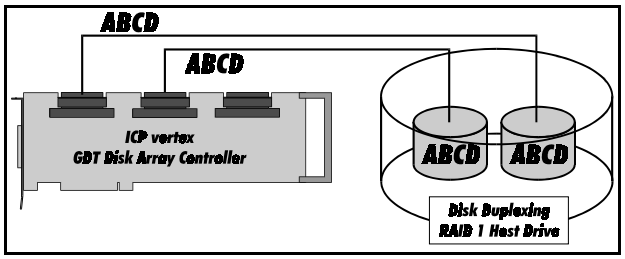
According to the adjusted stripe size (e.g., 16 KB) and the number of hard disks, the data blocks are split into stripes. Each stripe is stored on a separate hard disk. Especially with sequential read and write operations, we can observe a significant improvement of the data throughput. RAID 0 includes no redundancy at all, i.e., when one hard disk fails, all data is lost.



RAID 1 - Disk Mirroring/Disk Duplexing

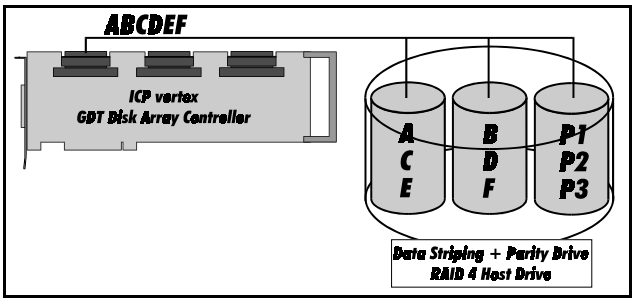


All data is stored twice on two identical hard disks. When one hard disk fails, all data are immediately available on the other without any impact on the performance and data integrity.
We talk about "Disk Mirroring" when two hard disks are mirrored on one SCSI channel. If each hard disk is connected with a separate SCSI channel, this is called "Disk Duplexing" (additional security).
RAID 1 represents an easy and highly efficient solution for data security and system availability. It is especially suitable for installations which are not too large (the capacity available is only half of the installed capacity).



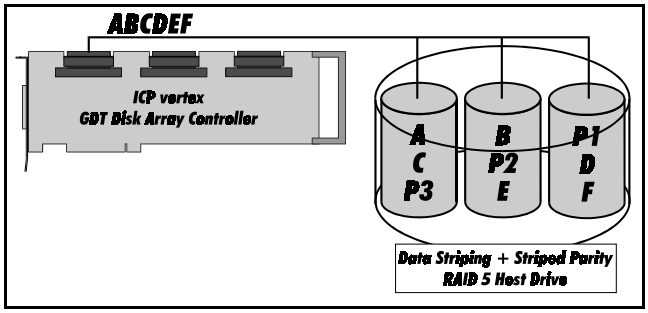
RAID 4 - Data Striping With a Dedicated Parity Drive

RAID 4 works in the same way as RAID 0. The data are striped amongst the hard disks. Additionally, the controller calculates redundancy data (parity information) which are stored on a separate hard disk (P1, P2, ...). Even when one hard disk fails, all data are still fully available. The missing data is recalculated from the data still available and the parity information. Unlike in RAID 1, only the capacity of one hard disk is needed for the redundancy. If we consider, for example, a RAID 4 disk array with 5 hard disks, 80% of the installed hard disk capacity is available as user capacity, only 20% is used for redundancy. In situations with many small data blocks, the parity hard disk becomes a throughput bottle-neck. With large data blocks, RAID 4 shows significantly improved performance.



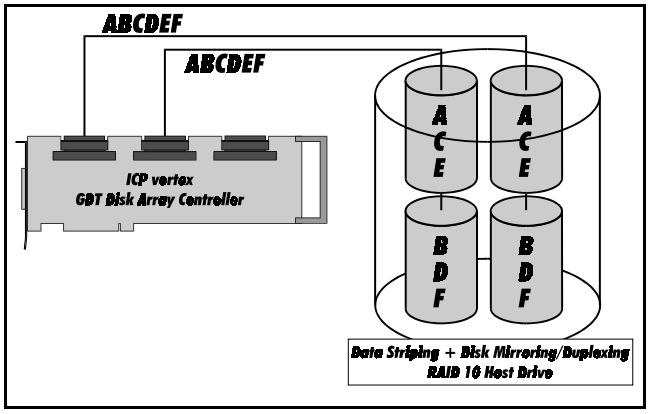
RAID 5 - Data Striping with Striped Parity

Unlike RAID 4, the parity data in a RAID 5 disk array are striped in all hard disks. The RAID 5 disk array delivers a balanced throughput. Even with small data blocks, which are very likely in a multi-tasking and multi-user environment, the response time is very good. RAID 5 offers the same level of security as RAID 4. When one hard disk fails, all data are still fully available, the missing data are recalculated from the data still available and the parity information. RAID 4 and RAID 5 are particularly suitable for systems with medium to large capacity requirements, due to their efficient ratio of the installed and actually available capacity.



RAID 10 - Combination of RAID 1 and RAID 0

The idea behind RAID 10 is simply based on the combination of RAID 0 (Performance) and RAID 1 (Data Security). Unlike RAID 4 and RAID 5, there is no need to calculate parity information. RAID 10 disk arrays offer good performance and data security. As in RAID 0, optimum performance is achieved in highly sequential load situations. Identical to RAID 1, 50% of the installed capacity is lost for redundancy.



C.3 How are the GDT Firmware Features Activated ?

Any installation or maintenance procedures regarding the ICP Controller are carried out with the configuration program **GDTSETUP**. The monitoring program **GDTMON** allows a continuous monitoring and maintenance of the ICP Controller and the connected disk arrays. The GDTMON utility also includes options to replace a defective drive with a new one (Hot Plug) and is available for most of the operating systems supported by the ICP Controllers. GDTSETUP allows you to set up single disks or complex disk arrays with simple and user-friendly installation procedures. Little previous knowledge is needed to be able to use GDTSETUP efficiently. It is only necessary to understand the hierarchy levels in the ICP Controller firmware (which are the same for both firmware versions: Standard and RAIDYNE). For the user's convenience the GDTSETUP program is available in two different variants:

- GDTSETUP loaded from the ICP Controller's Flash-RAM after switching on the computer
- GDTSETUP loaded from disk under MS-DOS.

The header of the GDTSETUP program indicates with a letter after the version number whether GDTSETUP was loaded from disk or from Flash-RAM:

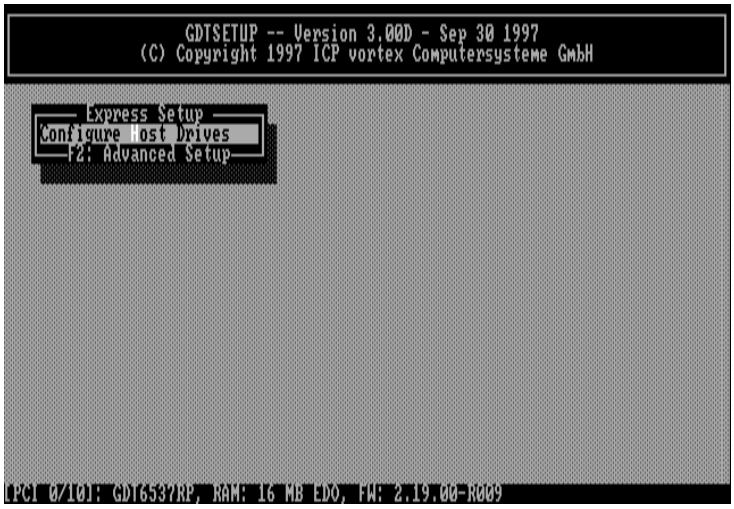
"**R**" for GDTSETUP loaded from the Flash-RAM after switching on the computer
"**D**" for GDTSETUP loaded from Disk, i.e., under MS-DOS.

Loading GDTSETUP with <CTRL><G> from the Flash-RAM is very comfortable since no operating system is required to carry out the configuration and setup works.

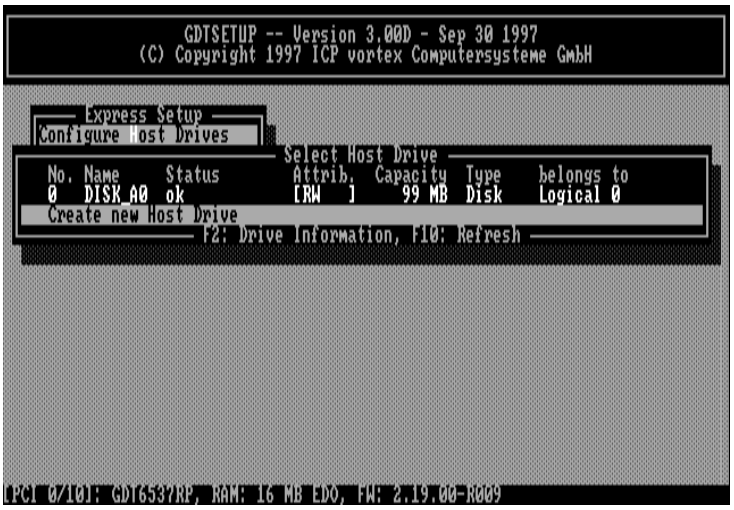
On the other side, loading GDTSETUP from disk (i.e., under MS-DOS) becomes necessary for tasks like partitioning or enabling a totally disabled GDT BIOS (which includes GDTSETUP).

C.3.1 The Express Setup Function of GDTSETUP

Whenever you load GDTSETUP and select the desired ICP Controller, it comes up in its EXPRESS Setup mode. This mode does not require any previous knowledge. If you choose this function, GDTSETUP carries out the complete installation entirely on its own, providing you for example with a fully operational RAID 5 Array Drive with optimized settings (for instance, with all SCSI features of a given drive activated).



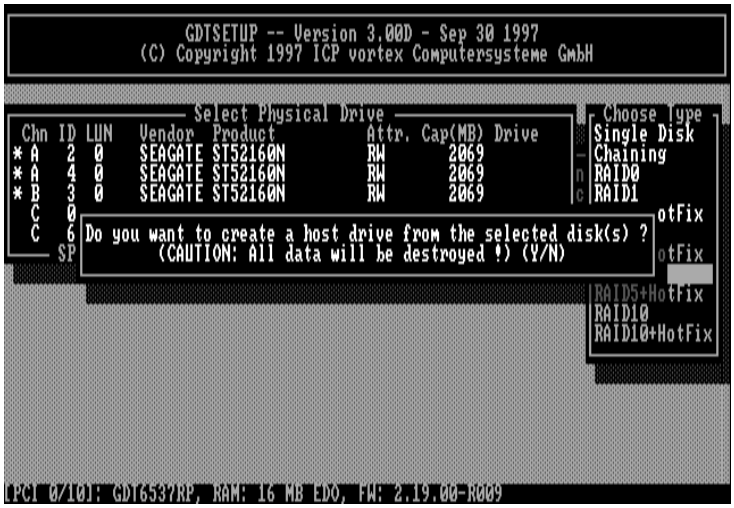
After selecting *Configure Host Drives*, select *Create new Host Drive*.



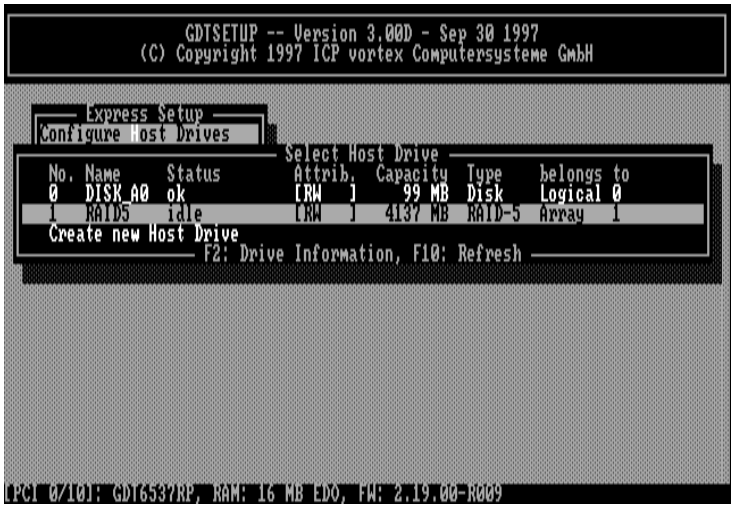
GDTSETUP scans the system for "free" hard disks (i.e., drives which are not yet part of other Host Drives). Use the <SPACE>-bar to select the desired hard disks (they are marked with an "*"). On the right side GDTSETUP offers highlighted the possible configurations with these drives.

Pressing <ENTER> ends the selection.

After choosing a configuration type for an Array Drive, GDTSETUP displays a security request.



After the confirmation, the Host Drive is automatically built up and configured.



After leaving GDTSETUP the parity information is generated.

For chapter C, we do not use this function, but give detailed instructions on how to set up a single disk and disk arrays with GDTSETUP and its *Enhanced Setup*.

C.4 Levels of Hierarchy Within the GDT Firmware

Both GDT firmware versions (Standard and RAIDYNE) are based on four fundamental levels of hierarchy. Each level has its "own drives" (= components). The basic rule is:

To build up a “drive“ on a given level of hierarchy, the “drives“ of the next lower level of hierarchy are used as components.

Level 1:
Physical Drives = hard disks, removable hard disks, some MO drives [1] are located on the lowest level. They are the basic components of all "drive constructions" you can set up. However, before they can be used by the firmware, these hard disks must be "prepared", a procedure we call *initialization*. During this initialization each hard disk receives information which allows a univocal identification even if the SCSI-ID or the controller is changed. For reasons of data coherency, this information is extremely important for any drive construction consisting of more than one physical drive.

Level 2:
On the next higher level are the **Logical Drives**. Logical Drives are introduced to obtain full independence of the physical coordinates of a physical device. This is necessary to easily change the whole ICP Controller and the channels, IDs, without loosing the data and the information on a specific disk array.

Level 3:
On this level of hierarchy, the firmware forms the **Array Drives**. Depending on the firmware installed, this can be

- Single Disks (one hard disk, some vendors call it JBOD - Just A Bunch Of Drives)
- Chaining Sets (concatenation of several hard disks)
- RAID 0 Array Drives
- RAID 1 Array Drives, RAID 1 Array Drives plus hot fix drive
- RAID 4 Array Drives, RAID 4 Array Drives plus hot fix drive
- RAID 5 Array Drives, RAID 5 Array Drives plus hot fix drive
- RAID 10 Array Drives, RAID 10 Array Drives plus hot fix drive

Level 4:
On the highest level of hierarchy, the firmware forms the **Host Drives**. In the end, only these Host Drives can be accessed by the host operating system of the computer. Drives C, D, etc. under MS-DOS, OS/2, etc. are always referred to as Host Drives by the firmware. The same applies to NetWare- and UNIX-drives. The firmware automatically transforms each newly installed Logical Drive and Array Drive into a Host Drive. This Host Drive is then assigned a Host Drive number which is identical to its Logical Drive or Array Drive number.
The firmware is capable of running several Host Drives of the most various kinds at the same time. An example for MS-DOS: drive C is a RAID 5 type Host Drive (consisting of 5 SCSI hard disks), drive D is a single hard disk, and drive E is a CD-ROM communicating with RAIDYNE through corelSCSI and the GDT ASPI manager.
On this level the user may split an existing Array Drive into several Host Drives.

---

[1] Also see section C.5.

After a capacity expansion of a given Array Drive the added capacity appears as a new Host Drive on this level. It can be either used as a separate Host Drive, or merged with the first Host Drive of the Array Drive.

Within GDTSETUP, each level of hierarchy has its own special menu:

Level 1	⇨	Menu: ***Configure Physical Devices***
Level 2	⇨	Menu: ***Configure Logical Drives***
Level 3	⇨	Menu: ***Configure Array Drives***
Level 4	⇨	Menu: ***Configure Host Drives***

Generally, each installation procedure passes through these 4 menus, starting with level 1.

Therefore:

- First initialize the Physical Drives.
- Then configure the Logical Drives.
- Then configure the Array Drives (e.g. Array Drives with RAID 0, 1, 4, 5 and 10).
- Finally, configure the Host Drives.

C.5 Using CD-ROMs, DATs, Tapes, etc.

A SCSI device that is not a SCSI hard disk or a removable hard disk, or that does not behave like one, is called a ***Not Direct Access Device***.
Such a device is **not configured with GDTSETUP and does not form a Logical or Host Drive**. SCSI devices of this kind are either operated through the ASPI interface (Advanced SCSI programming Interface) (MS-DOS, Windows, Novell NetWare or OS/2), or are directly accessed from the operating system (UNIX, Windows NT). For more information on how to use these devices, please refer to the corresponding chapters of this manual. Note: hard disks and removable hard disks are called *Direct Access Devices*. However, there are some *Not Direct Access Devices*, for instance certain MO drives, which can be operated just like removable hard disks if they have been appropriately configured before (for example by changing their jumper setting).

But enough on the dry theory. Now here are the examples which explain step by step all the necessary basics for setting up Host Drives with your ICP Controller

C.6 Example 1 - Installing a Single SCSI Hard Disk

This example is applicable to all ICP Controllers.
We presume that the controller and the SCSI hard disks have been installed properly.

Step 1: Loading GDTSETUP

You can load GDTSETUP in two ways:

1. Press the <CTRL><G> key combination when the GDT BIOS message comes up (shortly after switching on the computer) and load GDTSETUP from the Flash-RAM of the ICP Controller. In this case no operating system is required.

If GDTSETUP was loaded this way, there is an "R" (ROM) behind the version number.

2. Load GDTSETUP from disk under MS-DOS. Boot the MS-DOS-operating system (either from a boot-floppy or from an already existing boot drive, i.e., IDE-hard disk etc.). In order for GDTSETUP to work properly, you have to load the device driver GDTX000 first. This can be done in two ways:

   a.) Load GDTX000 from the DOS-command level by typing in GDTX000<ENTER>
   b.) Load GDTX000 automatically through the CONFIG.SYS file (DEVICE=GDTX000.EXE)

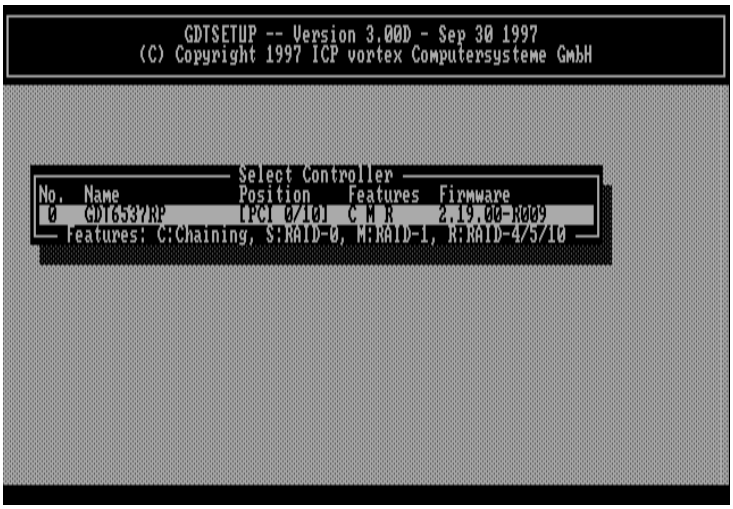
   Note: GDTSETUP.EXE as well as GDTX000.EXE are on the *System Disk - DOS*.

   If GDTSETUP was loaded this way, there is a "D" (Disk) behind the version number.

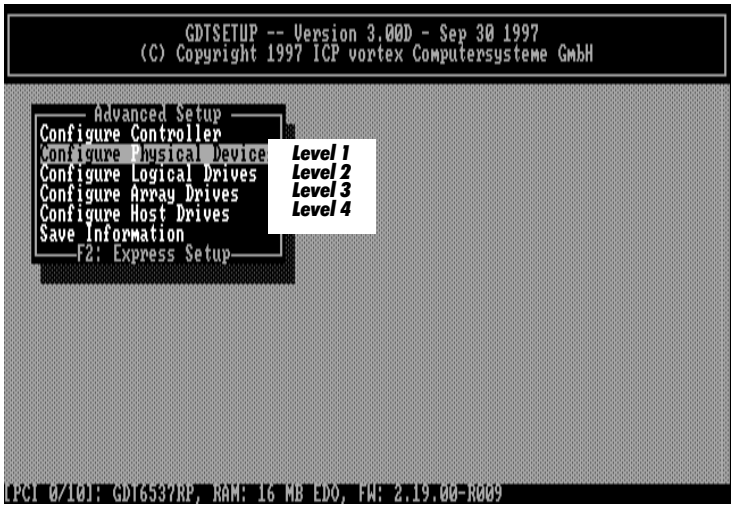
You may now ask what are the differences between the two GDTSETUP variants ?
They are small. The GDTSETUP variant loadable from disk under MS-DOS also additionally allows the partitioning of Host Drives, which is not possible with GDTSETUP loaded from the Flash-RAM. Loading GDTSETUP from the Flash-RAM is pretty easy, since there is nothing more required to configure the disk arrays. User's, who have for instance, an NT installation without a DOS partition, will highly appreciate this Flash-RAM-resident GDTSETUP.

For our example, it is not relevant whether we load GDTSETUP from the Flash-RAM, or from disk.

Now load GDTSETUP. The first menu asks you to select the desired ICP Controller. In our example, there is only one ICP Controller installed in the system. Therefore, simply press <ENTER> and then <F2> to select the Advanced Setup.



The main menu gives you the following options. As mentioned before, we have to go through levels 1 to 4 to install the SCSI hard disk (with almost nothing to do on levels 3 and 4).



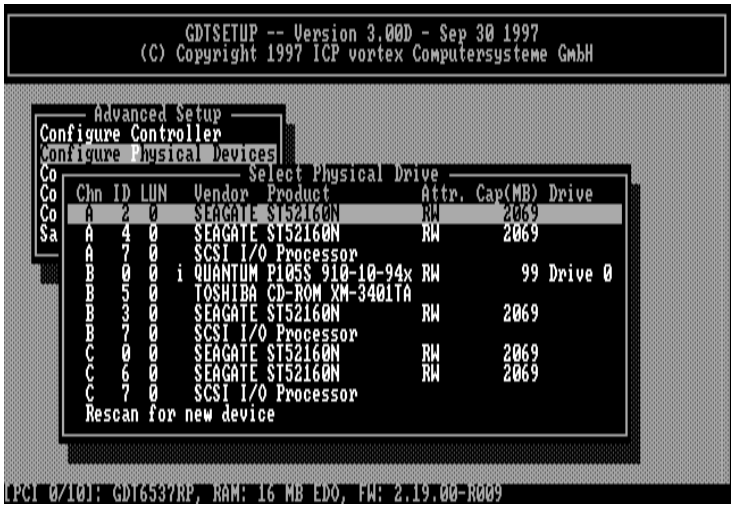
Step 2: Configure Physical Devices

Now activate the menu *Configure Physical Devices* (level 1). A list appears showing all hard disks found on the ICP Controller's SCSI channels. If you have a ICP Controller with a different number of SCSI channels, the existing SCSI channels are displayed. Note: This screen will always report all devices that are found to be connected to SCSI-cables, even though GDTSETUP only allows you to work on *Direct Access Devices* (and therefore not on tape drives, DATs, CD ROMs etc.).

The screen shows you:

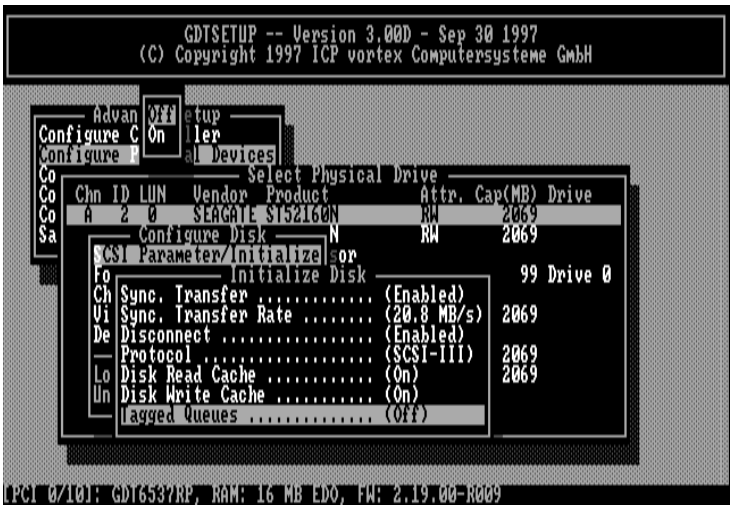
- the channel to which a SCSI device is connected
- which SCSI-ID the drive has (the entry SCSI I/O Processor stands for the corresponding SCSI channel of the ICP Controller. It has the default setting ID 7, as explained in chapter B)
- the initialization status
- the SCSI names of the drives
- the Read-Write-Status. [RW] = Read + Write
- the gross capacity
- membership in a Logical, Array or Host Drive

Use the cursor keys ↑ and ↓ to select the drive you wish to initialize. We take the first drive of channel A in the list. With this drive selected, press <ENTER>.



(Note: On Channel B, SCSI ID 0, is a drive which has been already initialized before. This is not relevant for our examples).

The *Configure Disk* menu appears which shows various options.
For our example we choose the *Initialize Disk* menu option and press <ENTER>.
The parameters within this menu can be changed by pressing <ENTER> and selecting the new setting.



1. Sync. Transfer: Enable
The SCSI-bus allows an asynchronous and a synchronous transfer. Every SCSI device must be able to perform the first type of transfer, the second one is optional. The advantage of the synchronous transfer lies in a higher data transfer rate as the signal transfer times on the possibly long SCSI-cable have no influence on the transfer rate anymore. Two SCSI-bus participants wanting to exchange data between each other have to check if and how (i.e., with which parameters) a synchronous data transfer between them is possible. Therefore, the mere setting does not automatically enable synchronous data transfer; this mode is only effective if both devices support it and after they have checked their capability of communicating with each other in this mode.

2. Sync. Transfer Rate
The maximum synchronous transfer rate can be limited. This limitation may become necessary if a particular SCSI cabling does not allow the maximum rate the drive and the controller could achieve. In our example, we leave the rate at 20.0 MB/s (for Wide SCSI at 20.0 MB/s and Wide & Ultra SCSI at 40.0 MB/s).
Note: In order to select a transfer rate above 10.0 MB/s the Protocol has to be set to SCSI-III.

3. Disconnect: Enable
The concept of the SCSI-bus allows several participants (8 IDs with 8 LUNs each). All these participants should be able to use the bus in a manner that causes the least reciprocal disturbance or obstruction. A participant should therefore vacate the bus if he does not need it. For reasons of performance, it is particularly important to guarantee a high degree of overlapping of the actions on the SCSI-bus. This high degree of overlapping can be achieved if a SCSI device is allowed to disconnect, thus leaving the bus to be used by other participants. If there is only one SCSI device connected to the SCSI-bus, Disconnect should be disabled.

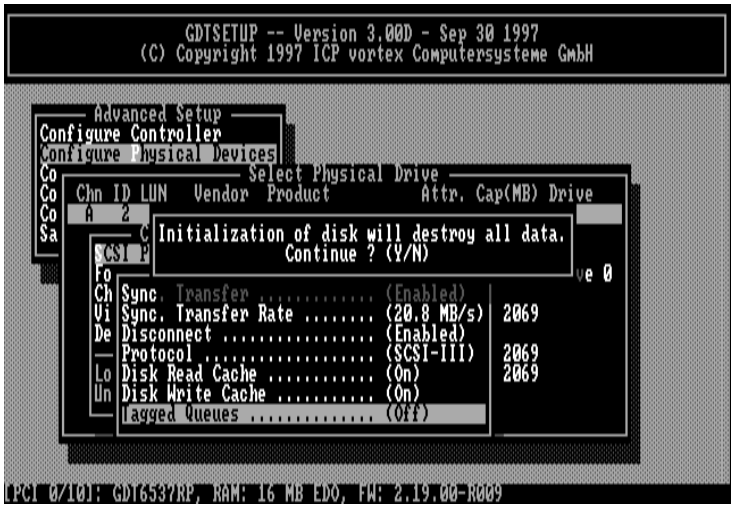
4. Protocol
This can be either SCSI-II or SCSI-III.
If you select SCSI-III make sure, that your hard disk supports this protocol. Most new multi-GB hard disks support SCSI-III. To enable Ultra (FAST-20) transfer rates (Narrow: 20 MB/s; Wide: 40 MB/s) SCSI-III protocol is required.

5. Disk Read Cache / Disk Write Cache / Tagged Queues
If a drive supports particular SCSI features you enable them (On).
Note: Most of the modern drives support disk caching (read and write). Some do not support Tagged Queues.

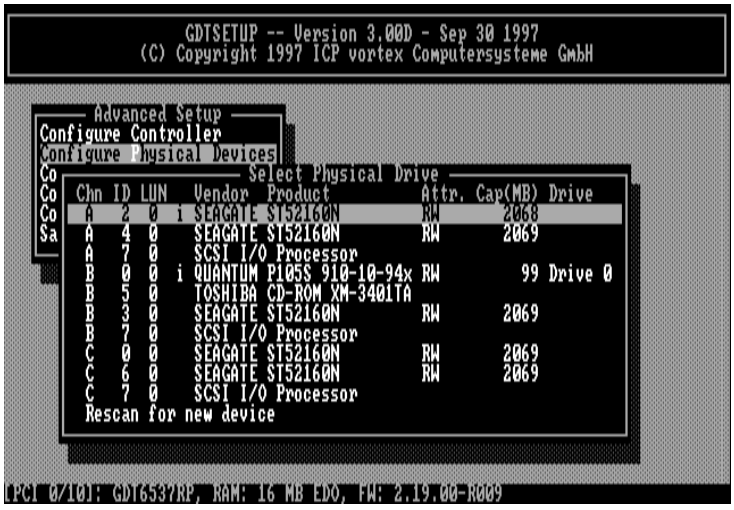
Press <ESC> to leave the *Initialize Disk* menu.
GDTSETUP displays a warning on the destruction of all data. This implies two different evaluations, according to the drive's current state and the options you have selected:

1. *First Initialization of the SCSI Device.*
   In this case the warning must be taken seriously. If the drive was previously connected to a different controller (e.g., NCR etc.) and still contains data, this data will be lost now.

2. *The SCSI Device was already initialized.*
   If only internal parameters, such as Disconnect, Synchronous Transfer and SCSI-II options have been changed, the data on the drive remains intact. Only the function state of the device is changed.

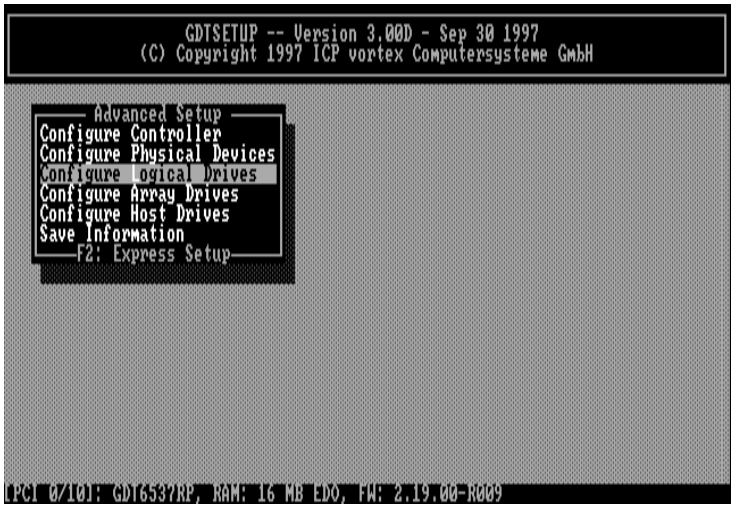


Press <Y> and we are back on the main screen of level 1 and see that the initialization-status of the SCSI device has changed.

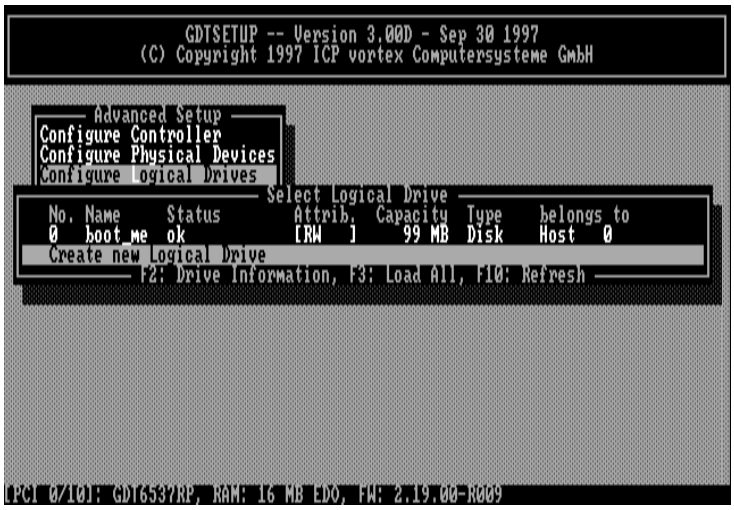


Step 3: Configure Logical Drives

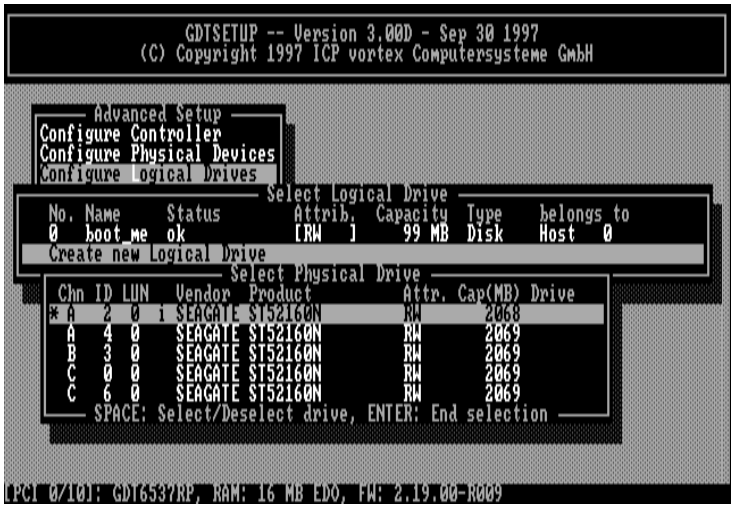
We now leave level 1 (by pressing the <ESC>-key) and are back in the main menu. Now, with the cursor keys ↑ and ↓ select *Configure Logical Drives* and go to level 2 by pressing <ENTER>.



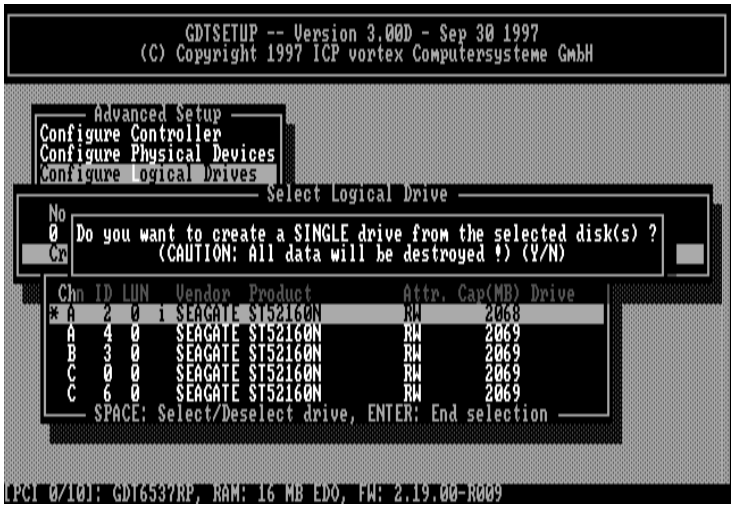
The main screen of level 2 appears. Move the selection bar to *Create new Logical Drive* and press <ENTER> .



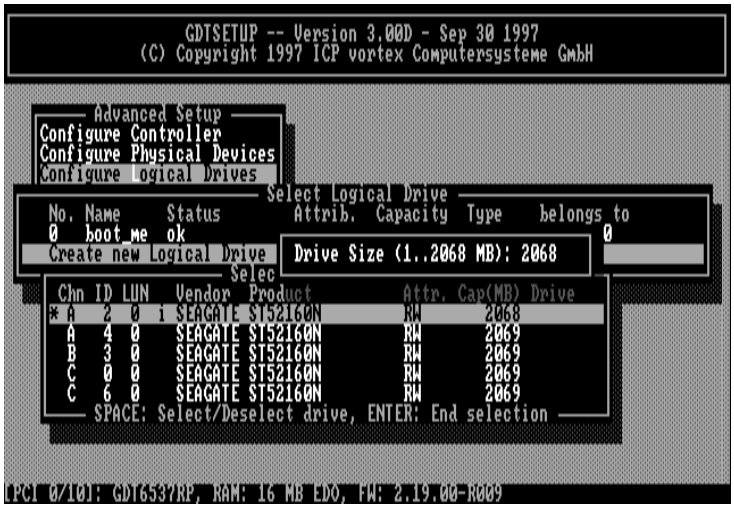
Note: The already existing Logical Drive in this list has no relevance for our example.



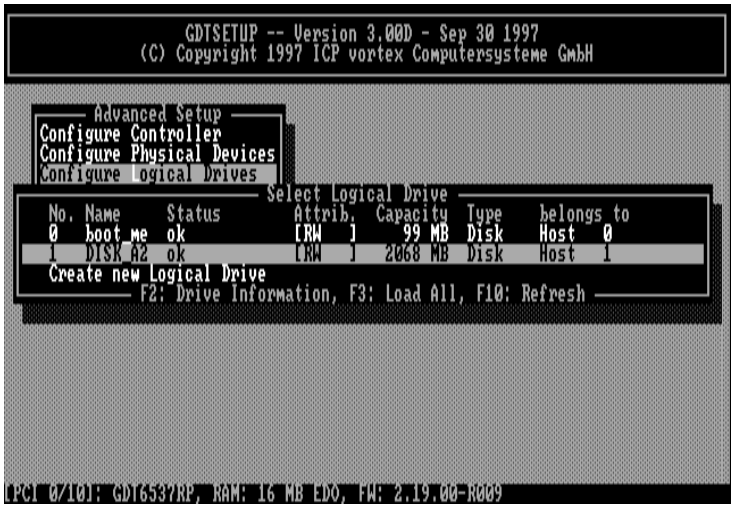
Select the initialized hard disk with the <SPACE>-bar (it becomes marked with an "*") and press <ENTER>.



For security reasons, you will be asked again if you want to use the selected disk to create a Logical Drive.
As we are sure of our choice, we confirm with <Yes>. GDTSETUP allows you to limit the hard disk size for this Logical Drive. This becomes interesting when you configure disk arrays. For this example we use the full capacity and press <ENTER> .



The dialog box is closed and we are back in the main menu of level 2.



As you can see, we have already created a new Logical Drive of the type *Disk*. The name of the Logical Drive is assigned automatically and contains the channel description and the

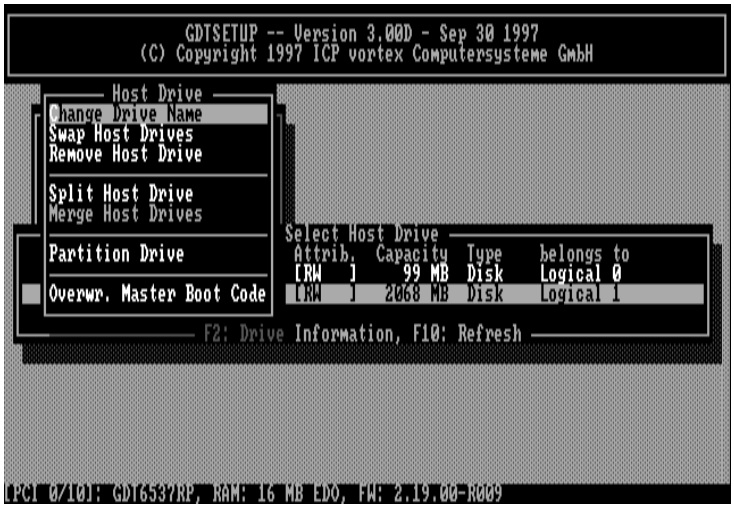
SCSI-ID after the "_" . This can serve as a reminder when you install a complex system with many drives. (Naturally, you may change the name.)
This concludes the installation on level 2. Now press the <ESC>-key to leave this screen.

Since we have only a single disk assigned to a Logical Drive, there is nothing to do in the *Configure Array Drives* menu, thus we go directly to the *Configure Host Drives* menu and have ***no Step 4.***

Step 5: Configure Host Drives

We are now back in the main menu of GDTSETUP and select *Configure Host Drive*.
The main screen of level 4 appears. Press <ENTER> . A list of available Host Drives is displayed. Again, the first entry is not relevant for our example.
At position 1 we find our previously configured Logical Drive. It was automatically transformed into a Host Drive, thus for this example we have nothing to do in this menu.

Press <ENTER> to get a list of possible menu options.



We should not forget to mention, that if you would have selected *Create new Host Drive*, this would have lead you to the same menu as the Express Setup mode.
But this example is an exercise which should help you to gain a better understanding of how the ICP Controller and GDTSETUP work. So don't believe we let you do redundant homework.
By the way, if you have loaded GDTSETUP from the Flash-RAM (<CTRL><G>) the *Partition Drive* option will be missing in this menu. The reason is that partitioning makes no sense when there is not an operating system loaded and the INT13H extension of the ICP Controller has not yet been activated.

Step 6: Leaving GDTSETUP

We are now back in the main menu of GDTSETUP. The installation is completed, and we therefore leave GDTSETUP by pressing the <ESC>-key. The following message appears:

As we are done with the installation and therefore definitely want to leave GDTSETUP, we press any key.

IMPORTANT: Always end GDTSETUP by leaving the program in the regular way (do not warm-boot with CTRL-ALT-DEL or cold boot by pressing the RESET button). Certain information is only transferred to the controller when you leave GDTSETUP in the regular way.

The Host Drive we have configured in this example is now ready for the installation of the desired operating system.

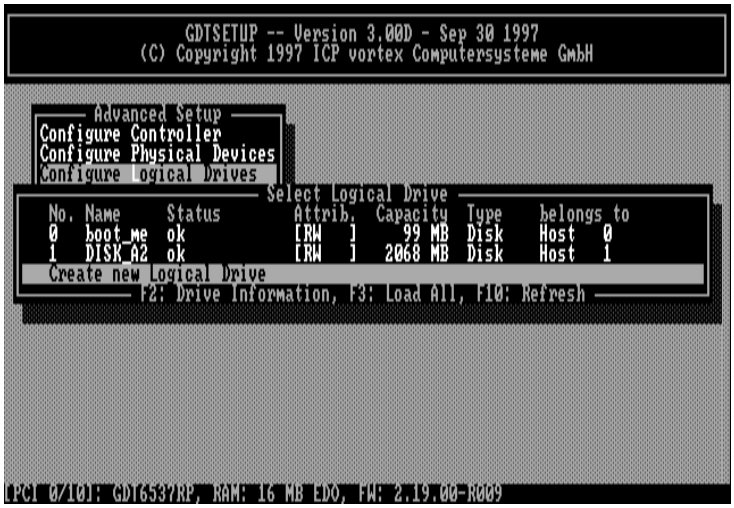
C.7 Example 2 - Installing a Mirroring Array - RAID 1

This example is applicable to all ICP Controllers.
It is our intention to install a Mirroring Array consisting of two identical hard disks. In the classical terminology of the RAID levels this is called a RAID 1 disk array.
We presume that the controller and the SCSI hard disks have been properly installed. **Step 1** of the installation is the same as in the first example, therefore we do not explain it again. **Step 2** regards the initialization of the second SCSI device. Proceed as described in the first example.

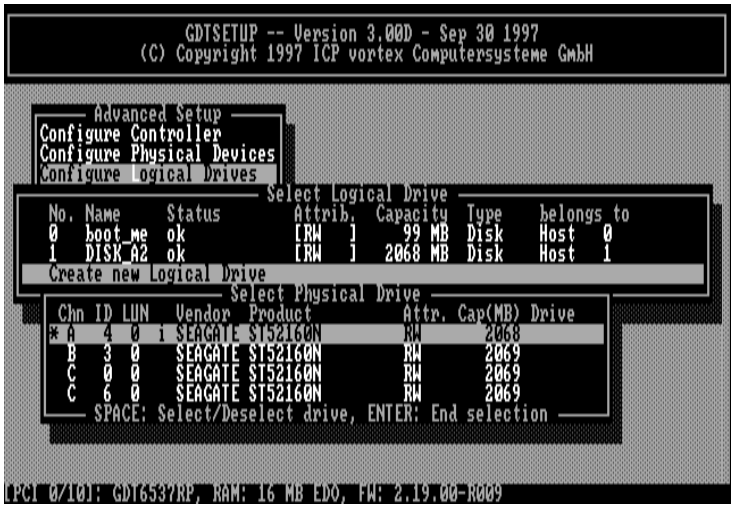
Step 3: Configure Logical Drives

We now leave level 1 (by pressing the <ESC>-key) and are back in the main menu. Now, with the cursor keys ↑ and ↓ select *Configure Logical Drives* and go to level 2 by pressing <ENTER>.

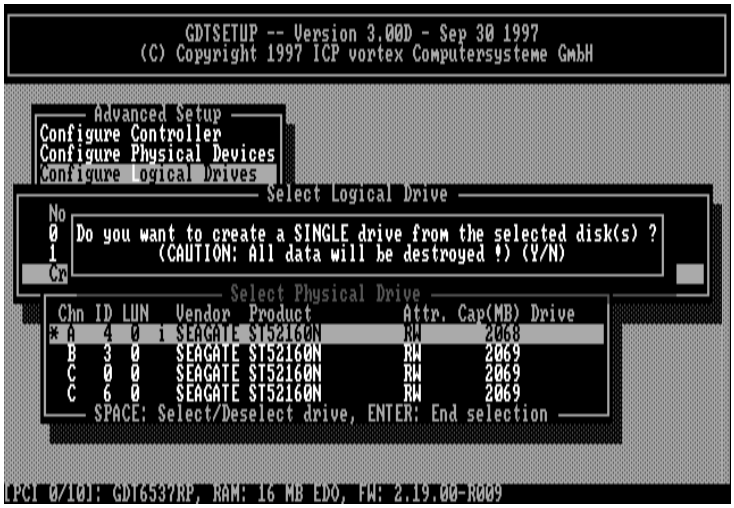
The main screen of level 2 appears. Move the selection bar to *Create new Logical Drive* and press <ENTER> .



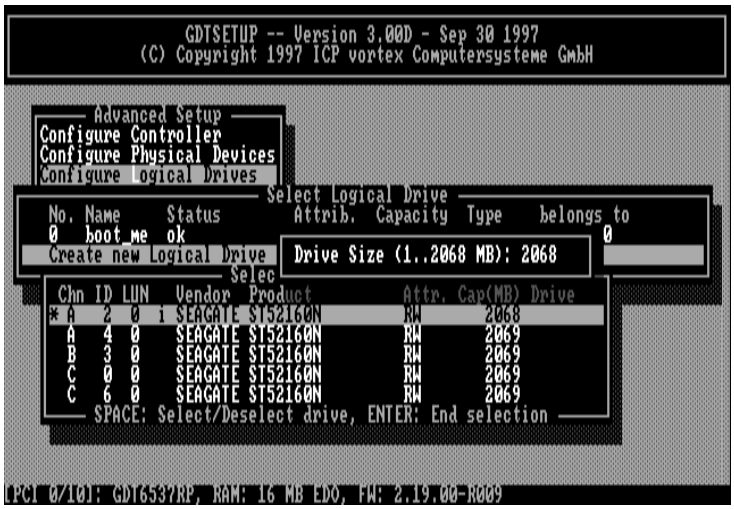
Note: The already existing Logical Drive in the first position of this list has no relevance for our example. The second entry was created before.



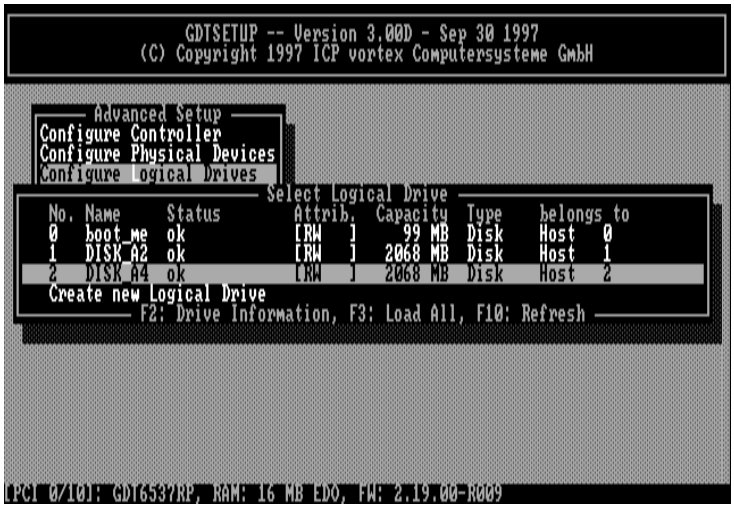
Select the initialized hard disk with the <SPACE>-bar (it becomes marked with an "*") and press <ENTER>.



For security reasons, you will be asked again if you want to use the selected disk to create a Logical Drive.
As we are sure of our choice, we confirm with <Yes>. GDTSETUP allows you to limit the hard disk size for this Logical Drive. This becomes interesting when you configure disk arrays. For this example we use the full capacity and press <ENTER> .



The dialog box is closed and we are back in the main menu of level 2.

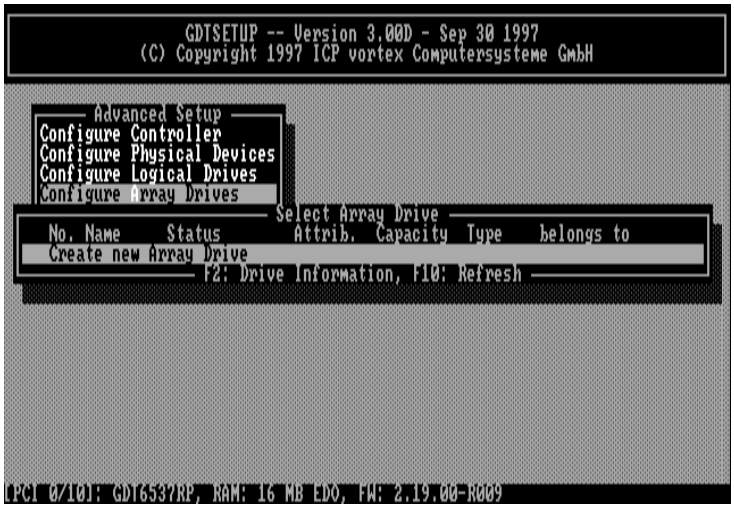


As you can see, we have created another Logical Drive of the type *Disk*. The name of the Logical Drive is assigned automatically and contains the channel description and the SCSI-ID after the "_" . This can serve as a reminder when you install a complex system with many drives. (Naturally, you may change the name.)
This concludes the installation on level 2. Now press the <ESC>-key to leave this screen.

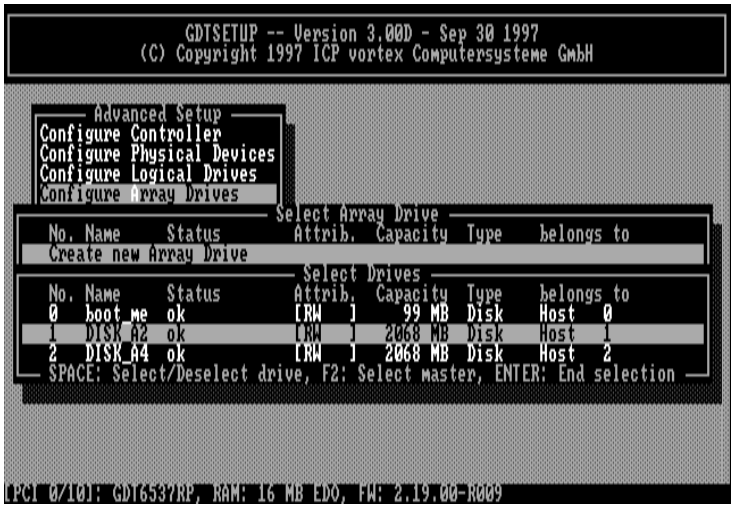
In the next step it is our objective to add the third Logical Drive in this list as a mirroring partner to the second Logical Drive of the list, thus configuring a RAID 1 disk array.

Step 4: Configure Array Drives

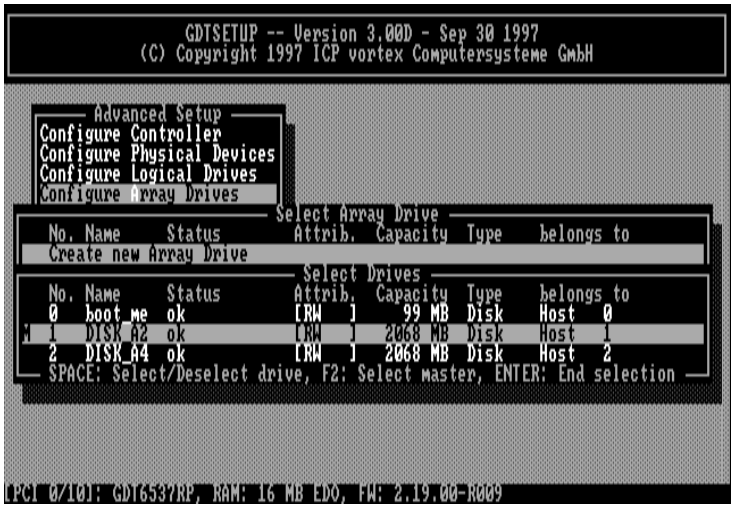
We now leave level 2 (by pressing the <ESC>-key) and are back in the main menu. Now, with the cursor keys ↑ and ↓ select *Configure Array Drives* and go to level 3 by pressing <ENTER>.



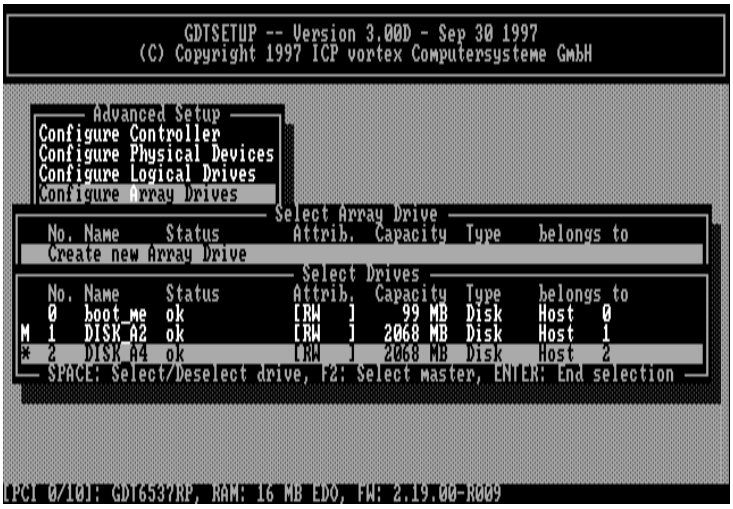
Since we want to create a new Array Drive press <ENTER>.
Note: The first entry in this list has no relevance for our example.



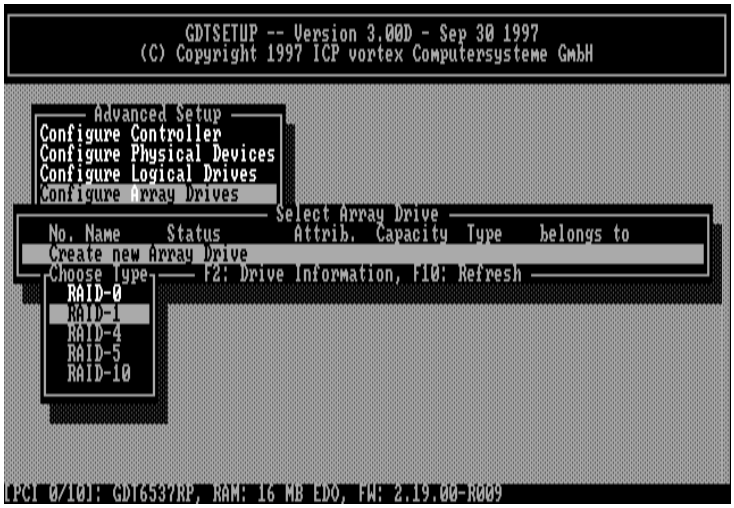
Move the selection bar to the second entry and press the <SPACE>-bar. The entry is marked with an "M" for Master. This means that the data from this Logical Drive are copied to the second Logical Drive, which we will select next.



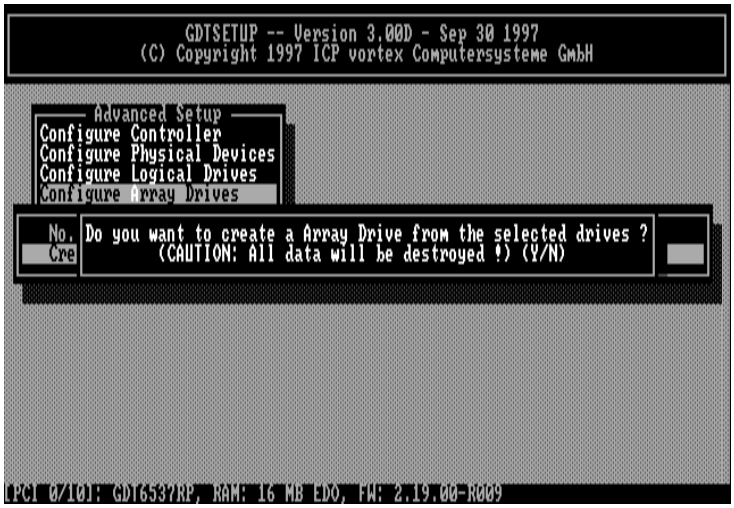
Move the selection bar with the cursor key ↓ to the next entry and press the <SPACE>-bar, again. It is marked with an "*"(pressing the <SPACE>-bar again undoes your choice).



When the Logical Drive is selected, confirm with <ENTER>. GDTSETUP displays now a list of possible RAID levels, available with the number of Logical Drives selected. In our case it is RAID-0 (data striping) and RAID 1. Move the selection bar to RAID-1 and press <ENTER>. (Note: RAID levels 4, 5 and 10 are only available with ICP Controllers which are equipped with the RAIDYNE firmware).

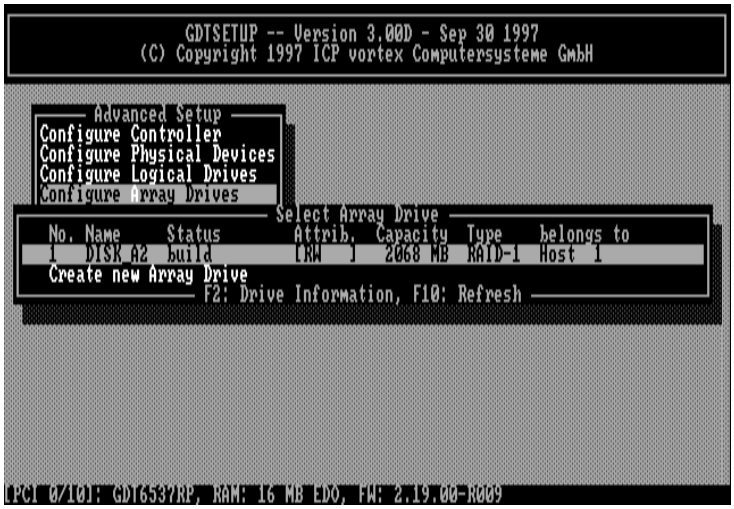


GDTSETUP displays a security request, which we answer with <Y>.



As you can easily recognize, we have created a new Array Drive of the Type RAID-1. Its state is build. When we leave GDTSETUP at the end of this example, you will see that the ICP Controller automatically copies the data of the first Logical Drive (our master) to the second Logical Drive. During this s*ynchronization* the RAID-1 array is fully operational.

The functioning of a RAID-1, or mirroring, disk array, is easy to understand: On the ICP Controller, one write-access from the host computer is transformed into two write-accesses (to both Logical Drives forming the mirroring array). If the two Logical Drives are built of hard disks, which are connected with different SCSI channels of the ICP Controller, both write-accesses are performed simultaneously (this method is often called *Disk Duplexing*). During a read-access of the host computer the data will be read from the Logical Drive whose hard disk has the fastest access to the data requested.
If a hard disk should fail (for instance due to a mechanical defect), all data is still available on the other Logical Drive. In this event, the controller gives an acoustical alarm.



Steps 5 and 6 are the same as in example 1.

C.8 Example 3 - Installing a RAID 5 Disk Array

This example is applicable to ICP Controllers with the RAIDYNE firmware.
The controller we use is the ICP Controller with three SCSI channels.

C.8.1 Mechanical Structure, Electrical Connections

Whether to install the SCSI hard disks into the computer case or into a separate disk subsystem enclosure strongly depends on your individual hardware equipment, therefore we shall not discuss it here. However, it is very important that the hard disks and the ICP Controller are cooled with a sufficient and constant air flow and that the power supply of the system is strong enough for the desired configuration. You may use a separate power supply for each hard disk in order to avoid power failure (what sense does a redundant Host Drive make if all hard disks forming the Host Drive are operated with one single power supply and this power supply fails ?). All participants of a SCSI-bus must have a different SCSI-ID. In addition, both ends of the SCSI-cable must be equipped with SCSI-bus terminator resistors. The SCSI-bus termination is crucial, since it is highly probable that a wrongly ter-

minated SCSI-bus will cause malfunctions of the connected devices and data transfer problems. The ICP Controller for this example has three independent SCSI channels: A, B, C. In our example, two SCSI hard disks are connected to channel A (DR1, DR2), one to channel B (DR3) and two to channel C (DR4, DR5). Our connection scheme is:

GDT-Channel A —— DR1 —— DR2
GDT-Channel B —— DR3
GDT-Channel C —— DR4 —— DR5

We make sure that all three channels have a proper SCSI bus termination. The SCSI-IDs are set according to the following list:

GDT channel A ——— ID 7 (default)
DR1 ——————————— ID 0
DR2 ——————————— ID 6

GDT channel B ——— ID 7 (default)
DR3 ——————————— ID 2

GDT channel C ——— ID 7 (default)
DR4 ——————————— ID 2
DR5 ——————————— ID 4

Also three SCSI-cables are needed. The cables for channel A and C have three connectors, where the cable for channel B has four, of which two connectors are used by a hard disk and a CD-ROM drive which are not relevant for our example.
Please note: bad SCSI-cables, wrong SCSI-IDs as well as a wrong termination of the busses are responsible for 95% of all possible errors !
In addition, it is essential that the hard disks and the controller be connected to the SCSI-cable with the right orientation. Although SCSI-cables are keyed, you should cross-check if all connectors of the cable have been pressed matching the correct key, especially when using home-made cables.
We recommend terminating the SCSI-cables at their ends opposite to the controller by means of so-called external terminator packs. These packs receive their terminator power directly from the cable. In this case, the termination is to be removed from or disabled on all hard disks. In order to get the best signal quality on the cable, the external terminator packs should have an active SCSI termination. Advantage: If you use the termination of a hard disk and this hard disk happens to fail in a manner that harms the bus termination, then it may occur that all devices connected to this cable do not function properly. Therefore, it is sensible to use external terminators for reasons of redundancy.

C.8.2 Setting up a RAID 5 Disk Array

Step 1: Loading GDTSETUP

You can load GDTSETUP in two ways:

1. Press the <CTRL><G> key combination when the GDT BIOS message comes up (shortly after switching on the computer) and load GDTSETUP from the Flash-RAM of the ICP Controller. In this case no operating system is required.
   If GDTSETUP was loaded this way, there is an "R" (ROM) behind the version number.

2. Load GDTSETUP from disk under MS-DOS. Boot the MS-DOS-operating system (either from a boot-floppy or from an already existing boot drive, i.e., IDE-hard disk etc.). In order for GDTSETUP to work properly, you have to load the device driver GDTX000 first.

This can be done in two ways:

a.) Load GDTX000 from the DOS-command level by typing in GDTX000<ENTER>
b.) Load GDTX000 automatically through the CONFIG.SYS file (DEVICE=GDTX000.EXE)

Note: GDTSETUP.EXE as well as GDTX000.EXE are on the *System Disk - DOS*.

If GDTSETUP was loaded this way, there is a "D" (Disk) behind the version number.

You may ask now, what are the differences between the two GDTSETUP variants ?
They are small. The GDTSETUP variant loadable from disk under MS-DOS also additionally allows the partitioning of Host Drives which is not possible with GDTSETUP loaded from the Flash-RAM. Loading GDTSETUP from the Flash-RAM is pretty comfortable, since there is nothing more required to configure the disk arrays. User's, who have for instance, an NT installation without a DOS partition, will highly appreciate this Flash-RAM-resident GDTSETUP.

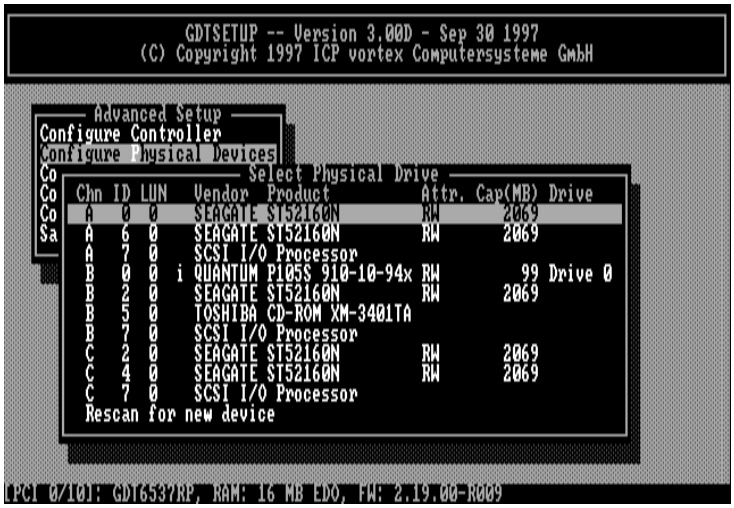
For our example, it is not relevant, whether we load GDTSETUP from the Flash-RAM, or from disk.

Now load GDTSETUP. The first menu asks you to select the desired ICP Controller. In our example there is only one ICP Controller installed in the system. Therefore simply press <ENTER> and then <F2> to select the Advanced Setup.

Step 2: Configure Physical Devices

Now activate the menu *Configure Physical Devices* (level 1). A list appears showing all hard disks found on the ICP Controller's SCSI channels. If you have a ICP Controller with a different number of SCSI channels, the existing SCSI channels are displayed. Note: This screen will always report all devices that are found to be connected to SCSI-cables, even though

GDTSETUP only allows you to work on *Direct Access Devices* (and therefore not on tape drives, DATs, CD ROMs etc.).



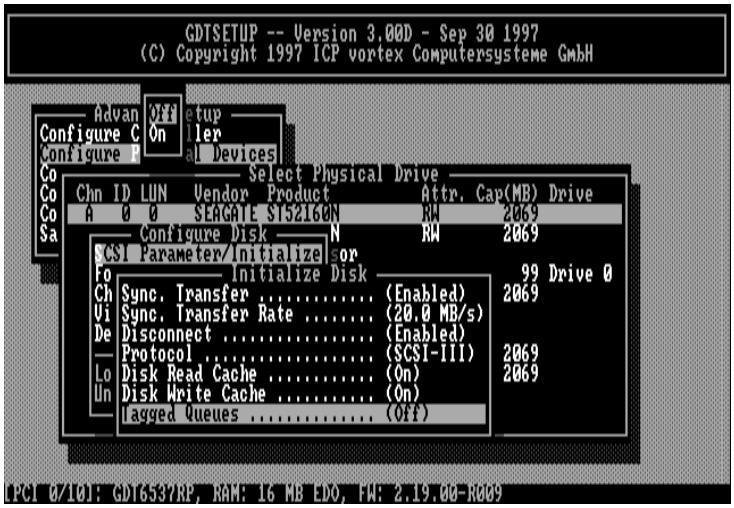
The screen shows you:

- the channel to which a SCSI device is connected
- which SCSI-ID the drive has (the entry SCSI I/O Processor stands for the corresponding SCSI channel of the ICP Controller. It has the default setting ID 7, as explained in chapter B)
- the initialization state
- the SCSI names of the drives
- the Read-Write-state. [RW] = Read + Write
- the gross capacity
- membership in a Logical, Array or Host Drive

Use the cursor keys ↑ and ↓ to select the drive you wish to initialize. We start with the first drive of the list. With this drive selected, press <ENTER>.

(Note: On Channel B, SCSI IDs 0 and 5, are devices which are not relevant for our example.)

The *Configure Disk* menu appears which shows various options.
For our example we choose the *Initialize Disk* menu option and press <ENTER>.
The parameters within this menu can be changed by pressing <ENTER> and selecting the new setting.



1. Sync. Transfer: Enable
The SCSI-bus allows an asynchronous and a synchronous transfer. Every SCSI device must be able to perform the first type of transfer, the second one is optional. The advantage of the synchronous transfer lies in a higher data transfer rate as the signal transfer times on the possibly long SCSI-cable have no influence on the transfer rate anymore. Two SCSI-bus participants wanting to exchange data between each other have to check if and how (i.e., with which parameters) a synchronous data transfer between them is possible. Therefore, the mere setting does not automatically enable synchronous data transfer; this mode is only effective if both devices support it and after they have checked their capability of communicating with each other in this mode.

2. Sync. Transfer Rate
The maximum synchronous transfer rate can be limited. This limitation may become necessary if a particular SCSI cabling does not allow the maximum rate the drive and the controller could achieve. In our example, we leave the rate at 20.0 MB/s (for Wide SCSI at 20.0 MB/s and Wide & Ultra SCSI at 40.0 MB/s).
Note: In order to select a transfer rate above 10.0 MB/s the Protocol has to be set to SCSI-III.

3. Disconnect: Enable
The concept of the SCSI-bus allows several participants (8 IDs with 8 LUNs each). All these participants should be able to use the bus in a manner that causes the least reciprocal disturbance or obstruction. A participant should therefore vacate the bus if he does not need it. For reasons of performance, it is particularly important to guarantee a high degree of overlapping of the actions on the SCSI-bus. This high degree of overlapping can be achieved if a SCSI device is allowed to disconnect, thus leaving the bus to be used by other participants. If there is only one SCSI device connected to the SCSI-bus, Disconnect should be disabled.

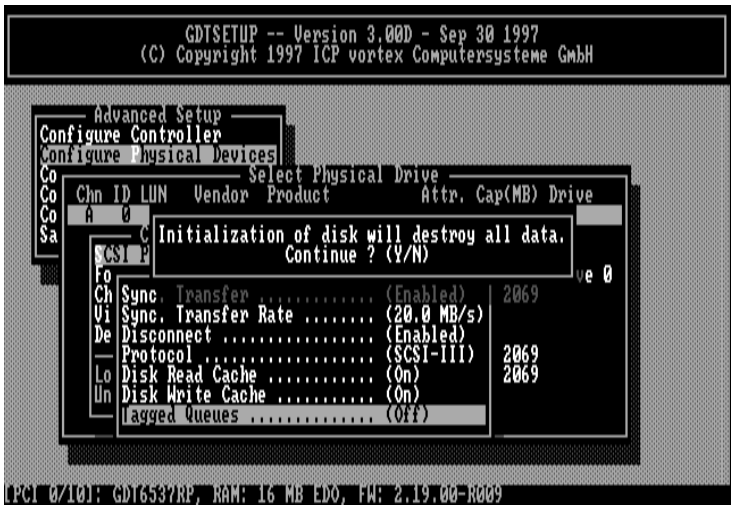
4. Protocol
This can be either SCSI-II or SCSI-III.
If you select SCSI-III make sure, that your hard disk supports this protocol. Most new multi-GB hard disks support SCSI-III. To enable Ultra (FAST-20) transfer rates (Narrow: 20 MB/s; Wide: 40 MB/s), SCSI-III protocol is required.

5. Disk Read Cache / Disk Write Cache / Tagged Queues
If a drive supports particular SCSI features you enable them (On).
Note: Most of the modern drives support disk caching (read and write). Some do not support Tagged Queues.

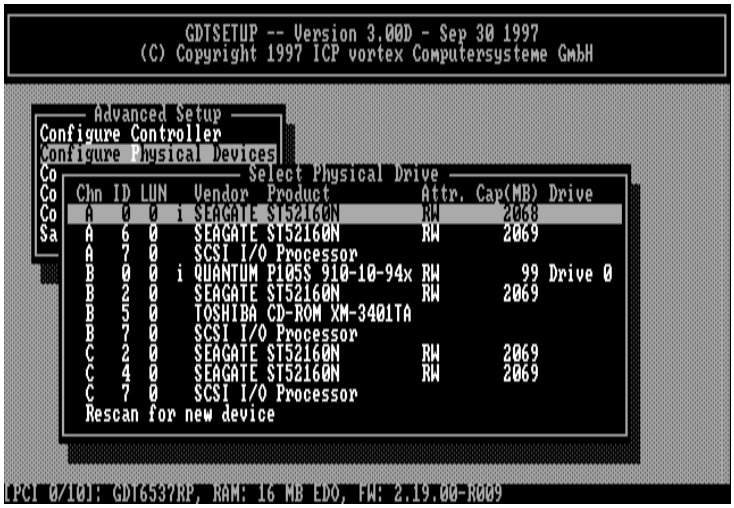
Press <ESC> to leave the *Initialize Disk* menu.
GDTSETUP displays a warning on the destruction of all data. This implies two different evaluations, according to the drive's current state and the options you have selected:

1. *First Initialization of the SCSI Device.*
   In this case the warning must be taken seriously. If the drive was previously connected to a different controller (e.g., NCR etc.) and still contains data, this data will be lost now.

2. *The SCSI Device was already initialized.*
   If only internal parameters, such as Disconnect, Synchronous Transfer and SCSI-II options have been changed, the data on the drive remains intact. Only the function state of the device is changed.



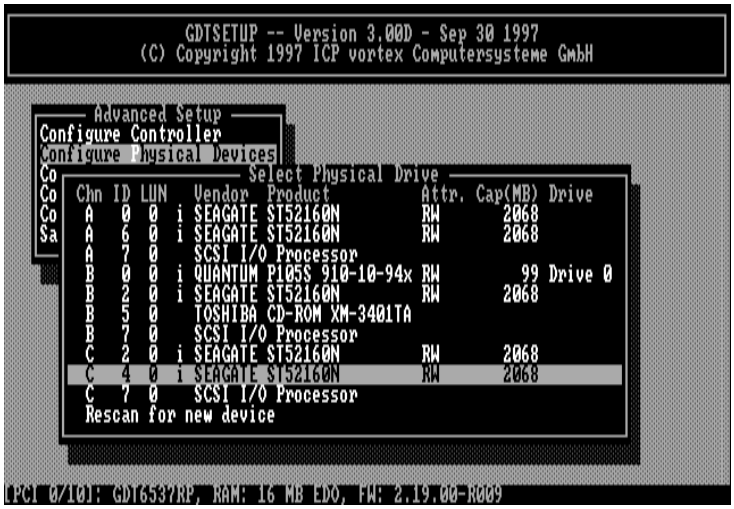
Press <Y> and we are back on the main screen of level 1 and see that the initialization-state of the SCSI device has changed.



Initialize the remaining four SEAGATE drives as described above, that is:

- Select the device with the cursor keys ↑ and ↓ and press the <ENTER>-key
- Choose the settings shown above
- Carry out the initialization

When the initialization of the last SCSI device has been completed, the screen should look as follows (a small **i** (i = initialized) must follow the SCSI-ID of each SCSI device):

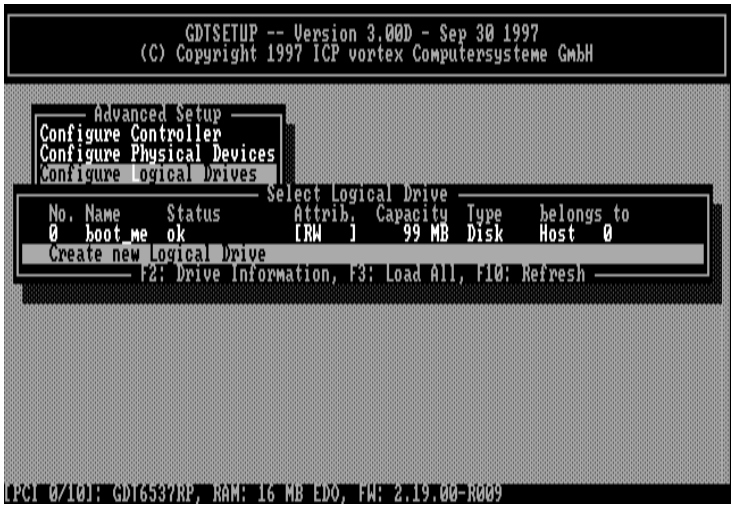


Important: Moving to the next level (Configure Logical Drives) only makes sense if all SCSI devices you need there are initialized.

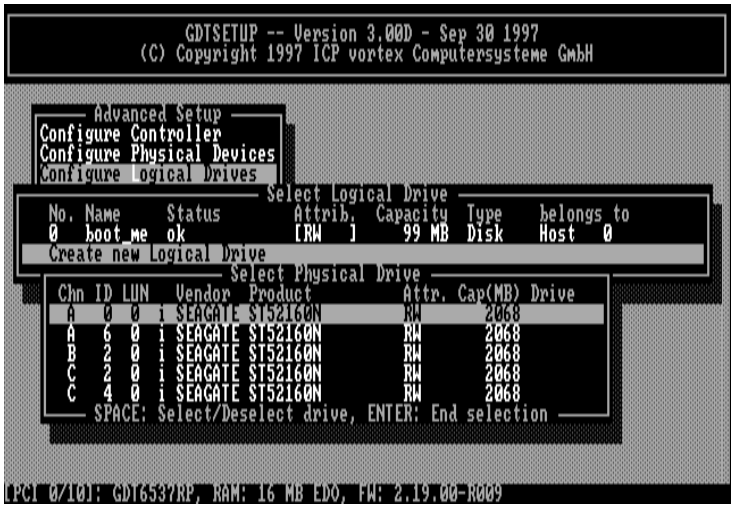
Step 3: Configure Logical Drives

We now leave level 1 (by pressing the <ESC>-key) and are back in the main menu. Now, with the cursor keys ↑ and ↓ select *Configure Logical Drives* and go to level 2 by pressing <ENTER>.

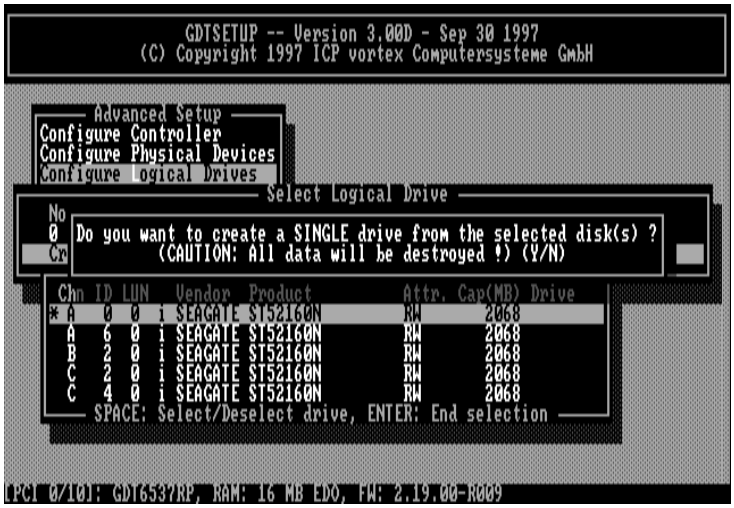
The main screen of level 2 appears. Move the selection bar to *Create new Logical Drives* and press <ENTER> .



Note: The already existing Logical Drive in this list has no relevance for our example.

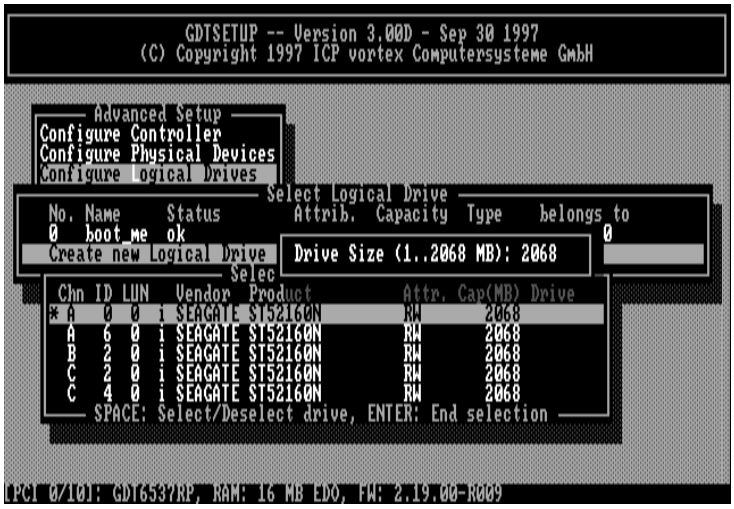


Select the initialized hard disk with the <SPACE>-bar (it becomes marked with an "*") and press <ENTER>.

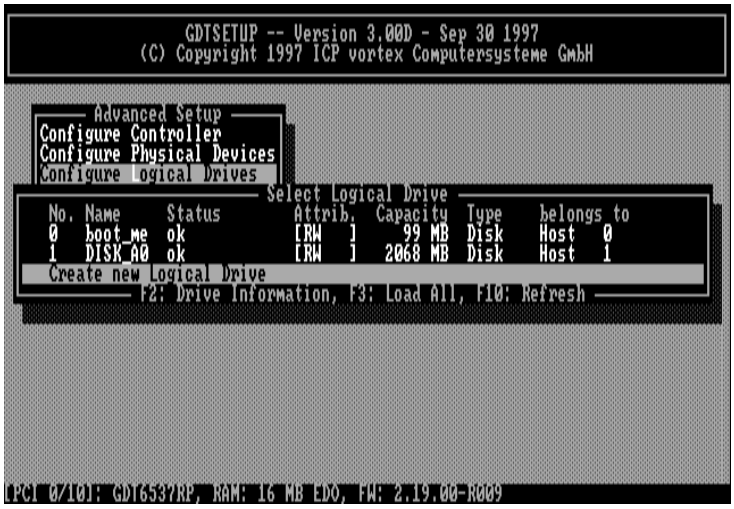


For security reasons, you will be asked again if you want to use the selected disk to create a Logical Drive. As we are sure of our choice, we confirm with <Yes>. GDTSETUP allows you to limit the hard disk size for this Logical Drive. This becomes interesting when you config-

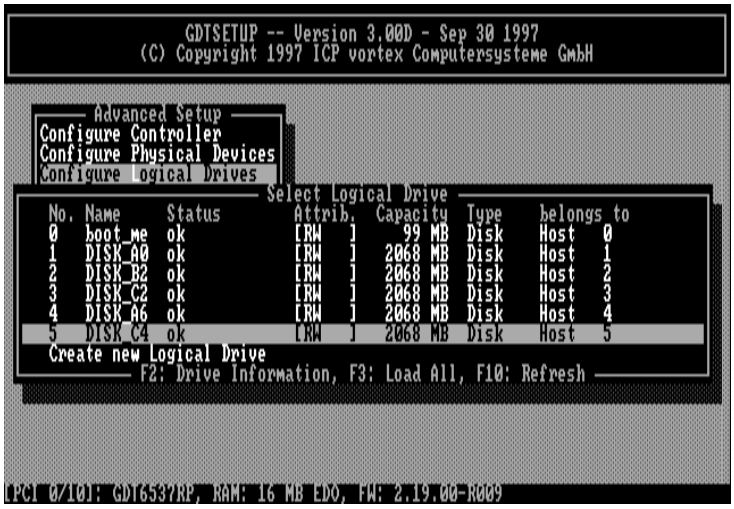
ure disk arrays and you want to make sure that future drives you want to bring into the disk array (e.g., for the capacity expansion or for replacement purposes) fit. It would be bad luck if the new drive only had 2067 MB. GDTSETUP couldn't accept it. To avoid this occuring, you could limit the capacity of each drive to 2000 MB. Any new 2 GB drive must have at least this capacity. The 68 MB in our example would be lost. For this example we use the full capacity and press <ENTER> .



The dialog box is closed and we are back in the main menu of level 2.



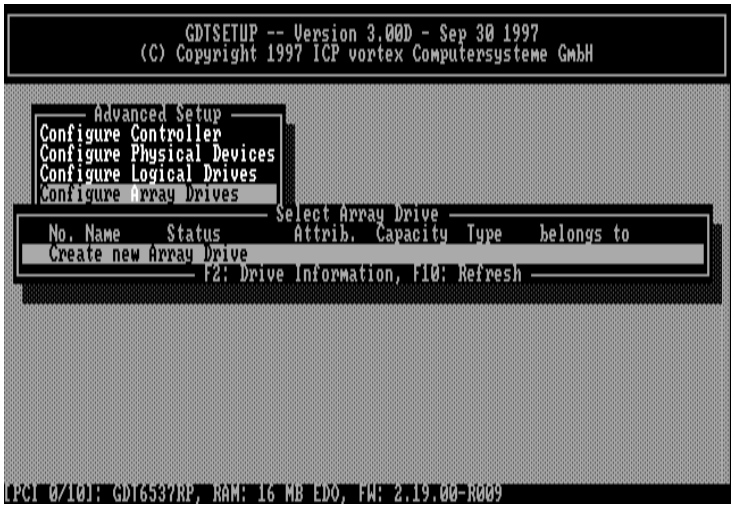
As you can see, we have already created a new Logical Drive of the type *Disk*. The name of the Logical Drive is assigned automatically and contains the channel description and the SCSI-ID after the "_" . This can serve as a reminder when you install a complex system with many drives. (Naturally, you may change the name.). Now configure the remaining Logical Drives one by one. Pay attention to choose the SCSI devices alternately from the SCSI channels. This selection method has a considerable impact on the disk array's performance because the data is written to the Logical Drives in stripes. If the consecutive Logical Drive is controlled by another SCSI channel, independent Logical Drive accesses become possible, resulting in a high degree of overlapping. After having completed these procedures for all five Logical Drives, you will see the following screen:



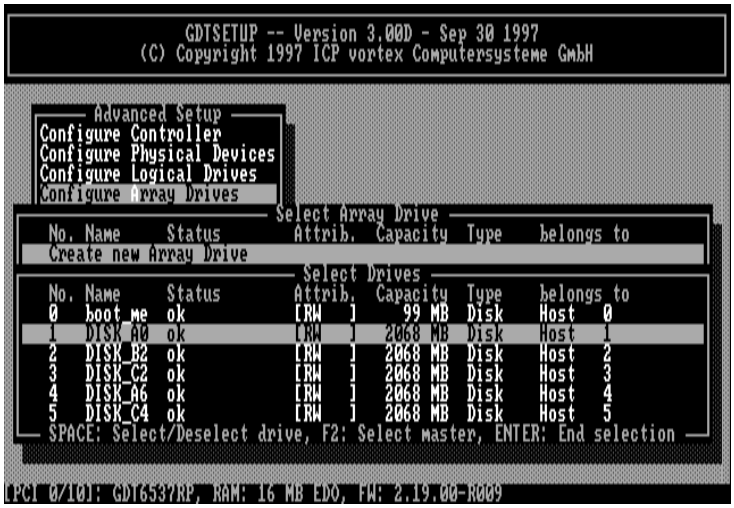
This concludes the installation on level 2. Now press the <ESC>-key to leave this screen.

Step 4: Configure Array Drives

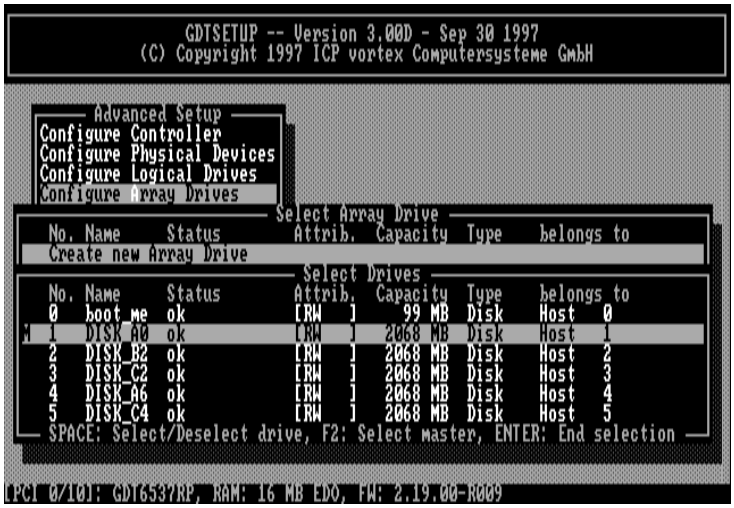
We now leave level 2 (by pressing the <ESC>-key) and are back in the main menu. Now, with the cursor keys ↑ and ↓ select *Configure Array Drives* and go to level 3 by pressing <ENTER>.



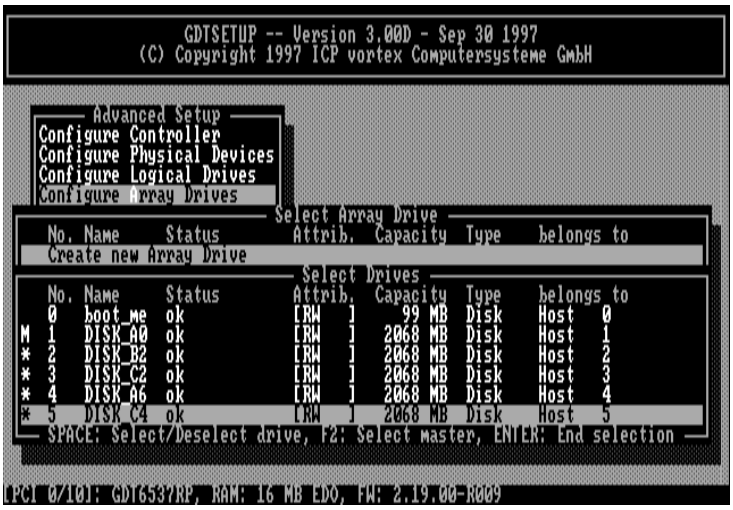
Since we want to create a new Array Drive press <ENTER>.
Note: The first entry in this list has no relevance for our example.



Move the selection bar to the second entry and press the <SPACE>-bar. The entry is marked with an "M" for Master. This means that the disk array "begins" with this Logical Drive.

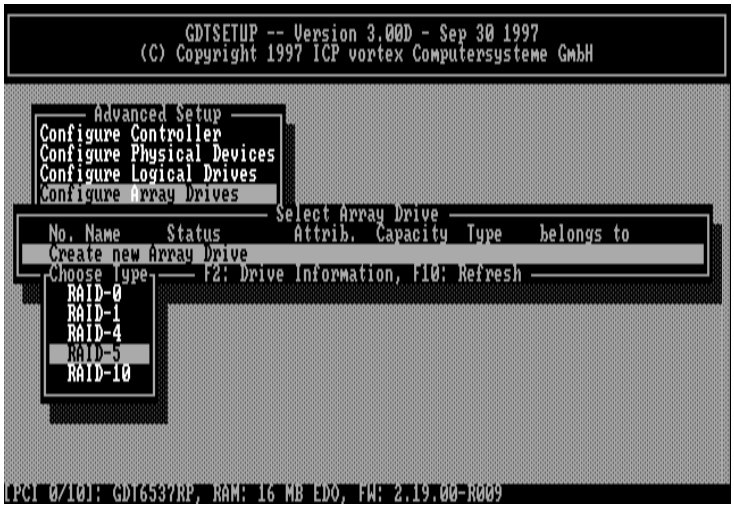


Move the selection bar with the cursor key ↓ to the next entry and press the <SPACE>-bar, again. It is marked with an "*"(pressing the <SPACE>-bar again undoes your choice). Repeat this selection until all five Logical Drives are marked.

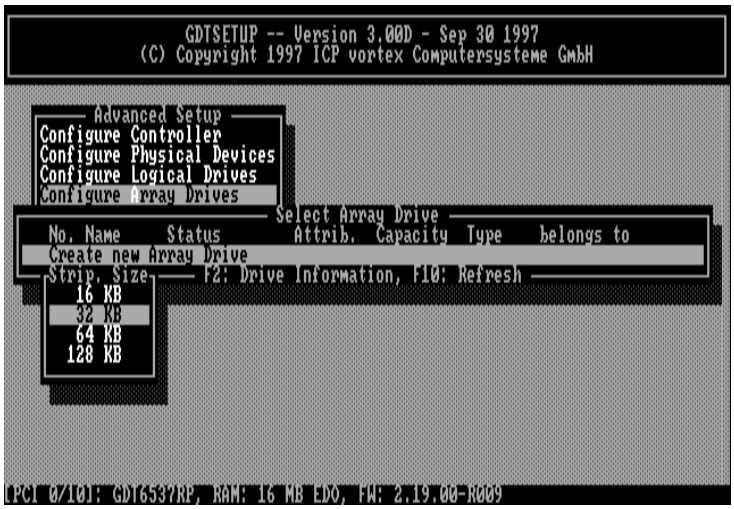


When the last Logical Drive is selected, confirm with <ENTER>. GDTSETUP now displays a list of possible RAID levels available with the number of Logical Drives selected.

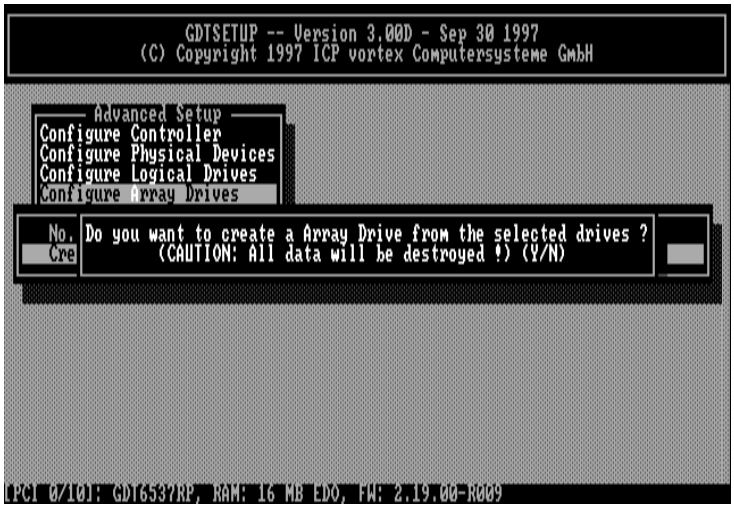
- RAID 0 pure data striping without redundancy
- RAID 1 disk mirroring
- RAID 4 data striping with dedicated parity drive
- RAID 5 data striping with striped parity
- RAID 10 RAID 0 combined with RAID 1



In our case we take RAID-5 and press <ENTER>.



GDTSETUP asks for the *Stripe Size*. This is the size of the stripes into which the data is divided. The default is 32KB which we leave for this example and therefore press <ENTER>. (Note: 32KB stripe size is suggested because in various performance tests it has proved to be the best value.). GDTSETUP displays a security request, which we confirm with <Y>.

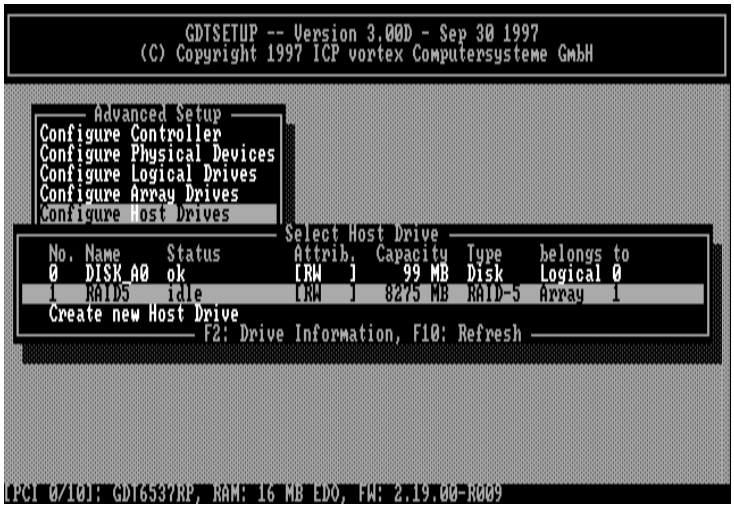


GDTSETUP allows you to limit the capacity of the disk array. This may be of interest if your installation requires an exact size for a disk array. Normally, the full capacity is used. In our example we press <ENTER> .

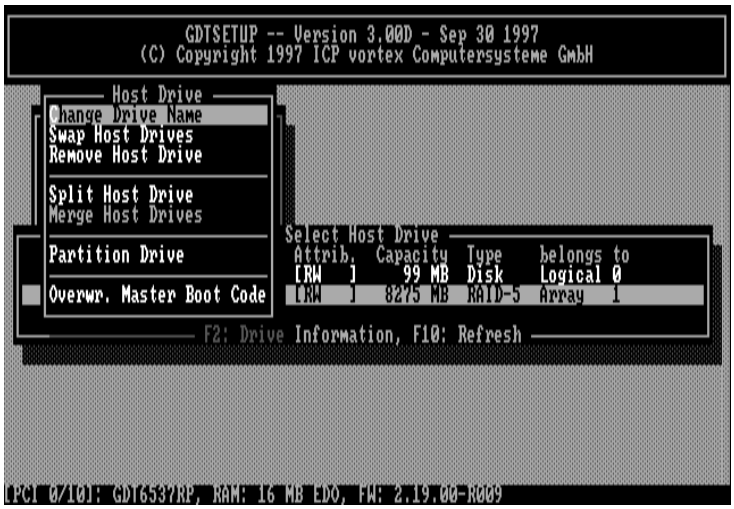
It's done !
We succeeded in setting up a RAID 5 disk array. The screen shows that the disk array is currently in an *idle* state. Later in this chapter, we shall explain the different states a RAIDYNE disk array can assume.
We are now back in the main menu of GDTSETUP.

Step 5: Configure Host Drives

We are now back in the main menu of GDTSETUP and select *Configure Host Drives*.
The main screen of level 4 appears. Press <ENTER> . A list of available Host Drives is displayed. Again, the first entry is not relevant for our example.
At position 1 we find our previously configured RAID-5 disk array. It was automatically transformed into a Host Drive, thus for this example we have nothing to do in this menu.



Press <ENTER> to get a list of possible menu options.



We should not forget to mention that if you would have selected *Create new Host Drive*, this would have lead you to exactly the same menu as the Express Setup mode.
But this example is an exercise which should help you to gain a better understanding of how the ICP Controller and GDTSETUP work.
By the way, if you have loaded GDTSETUP from the Flash-RAM (<CTRL><G>) the *Partition Drive* option will be missing in this menu. The reason is that partitioning makes no sense,

when there is no operating system loaded and the INT13H extension of the ICP Controller has not yet been activated.

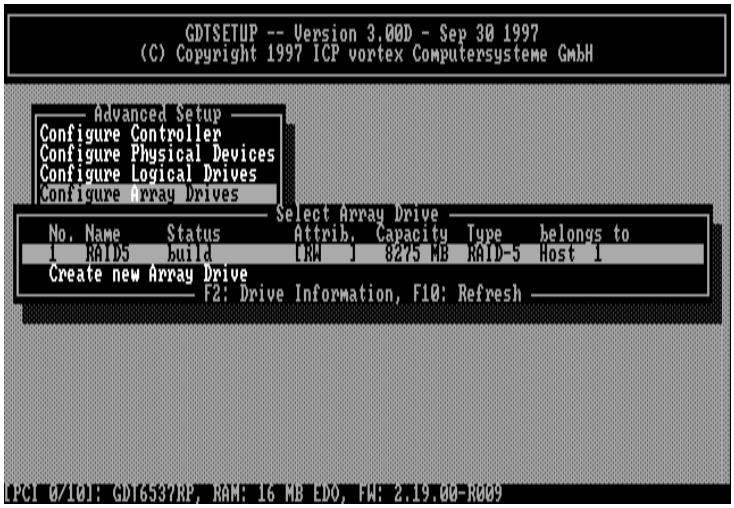
Step 6: Leaving GDTSETUP

We are now back in the main menu of GDTSETUP. The installation is completed, and we therefore leave GDTSETUP by pressing the <ESC>-key. The following message appears:

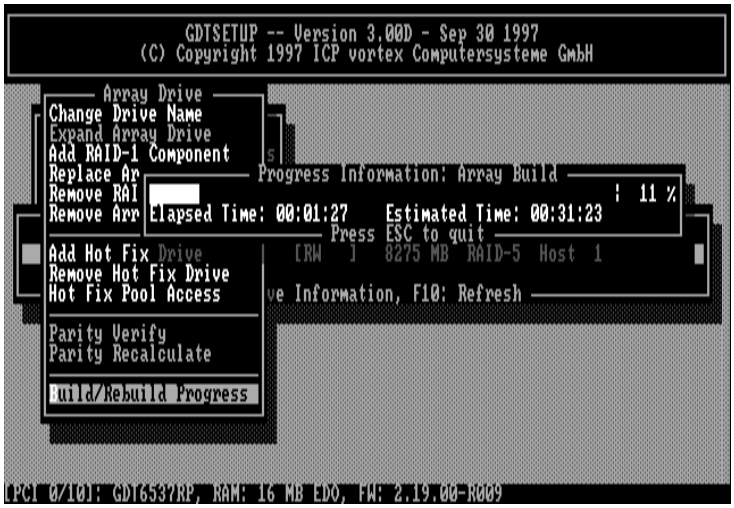
As we are done with the installation and therefore definitely want to leave GDTSETUP, we press any key.

IMPORTANT: Always end GDTSETUP by leaving the program in the regular way (do not warm-boot with CTRL-ALT-DEL or cold boot by pressing the RESET button). Certain information is only transferred to the controller when you leave GDTSETUP in the regular way.

After rebooting the system, load GDTSETUP anew. Change to the Advanced Setup menu, select *Configure Array Drives*, and press <ENTER> . As you can see, the disk arrays Status has changed to build, i.e., the parity information is currently generated and written to the disks (you may already have noticed the disk activity).

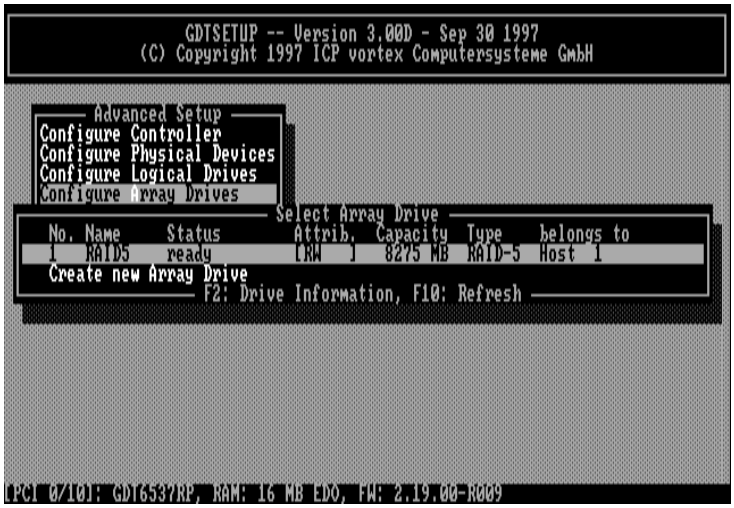


Press <ENTER> and move the selection bar to the *Build/Rebuild Progress* menu. Press <ENTER> .

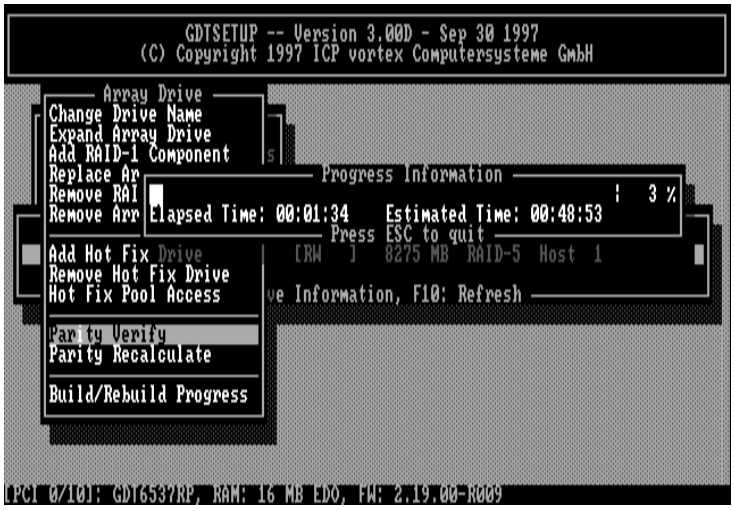


From the progress information slider, we can easily see, that the 8 GB disk array is already built up 11% and that the estimated time for the build process is 31 minutes.
Note: During the build process the disk array is fully operational, but not yet redundant. I.e., you could immediately start installing your desired operating system, without having to wait until this process has finished.

At the end of this build process the disk arrays state becomes ready (fully redundant).



Now press <ENTER> and move the selection bar to the *Parity Verify* menu. Press <ENTER>.



RAIDYNE now checks the correctness of the redundancy information (i.e., calculates the redundancy information anew and compares it with the already existing information).

Depending on how large the disk array is, this check may take quite a long time, however, it can be aborted by pressing <ESC>. *Parity Verify* is a diagnosis function which enables you to verify the consistency of a disk array every now and then. We interrupt the verification by pressing <ESC>. Note: The GDT monitor program **GDTMON** also includes the parity verify function. Unlike in GDTSETUP, the disk array's parity can be checked while the disk array is fully operational (e.g., in a NetWare file server). Further information on GDTMON is given in a separate chapter of this manual.

Step 7: Simulating a Drive Failure

This part of our example is optional. Nevertheless, we recommend that you go through it. It gives you a better understanding of how RAIDYNE reacts in the event of a drive failure and what you have to do in such a case.

Important: To carry out the drive failure simulation, the disk array must be in the *ready* state. Only in this state, has the disk array redundancy.

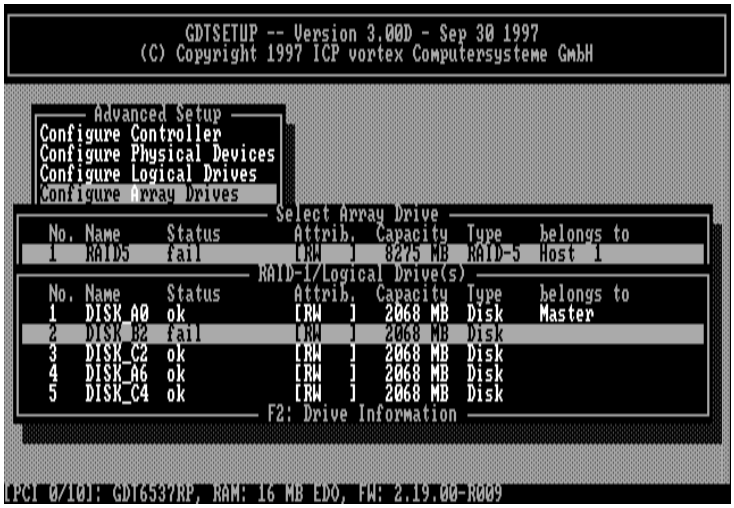
In order to make the simulation as realistic as possible, we suggest creating an MS-DOS-partition on the disk array with FDISK. To create disk activity, write a small batch program which copies data from one directory of this partition to another. While the copy process is going on, we simulate a drive failure of drive DISK_B2 by removing the drive's DC power supply cable. (If you choose to let another drive fail, please keep in mind the information on the SCSI-bus termination given in *Step 1*). Now we can observe how RAIDYNE reacts:

1. After a short time, the acoustical alarm of the GDT is activated. (Note: this alarm is only activated when the RAID 5 Array Drive is being accessed).

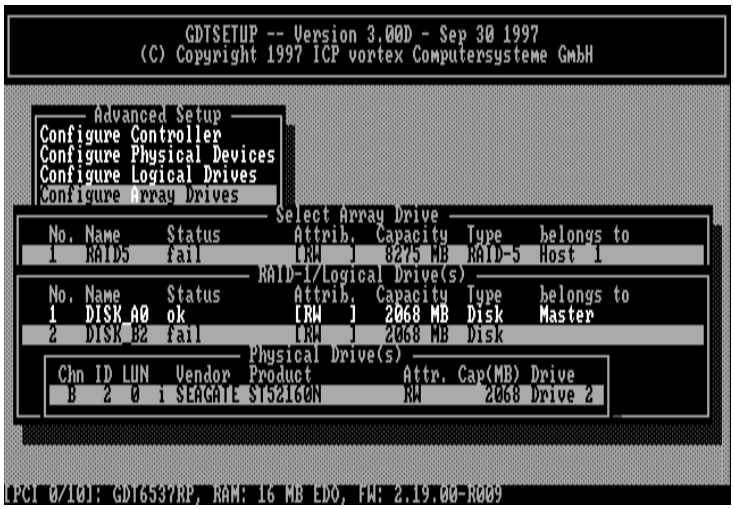
2. RAIDYNE activates the so-called *fail* operation state during which the disk array remains fully operational. The data of the failed drive is calculated by means of the redundant data stored on the other drives.

The alarm signal does not switch off because the disk array, although operational, is found in a state without redundancy, that is, a state which should be eliminated as soon as possible. The alarm signal turns off when GDTSETUP is loaded.
How is this situation reflected in GDTSETUP ?
What has happened to the failed hard disk ?

To answer these questions, we load GDTSETUP and check. We go to the menu *Configure Array Drives* menu and select our RAID-5 disk array which entered the *fail* state. Press <F2> to get further information on the failure.

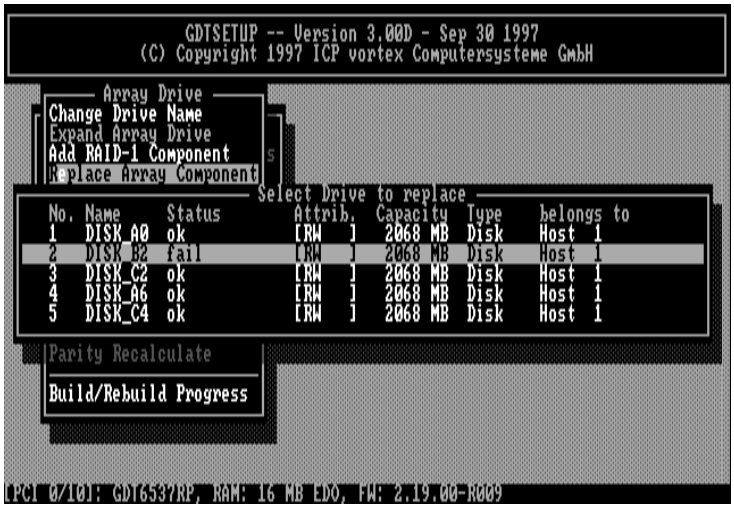


After selecting the failed Logical Drive, press again <F2> to obtain detailed information on the physical drive which has actually failed.

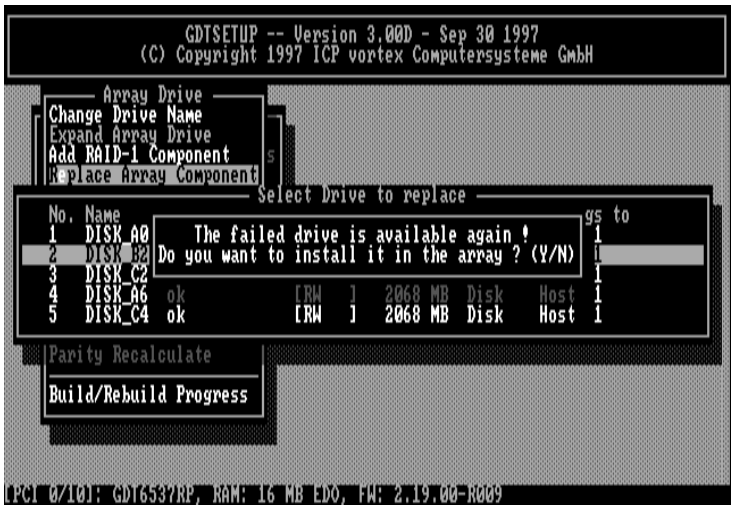


Important: Even if we reconnected the power supply to DISK_B2 before loading GDTSETUP, DISK_B2 would not be included in the disk array again. If you decide to use the failed hard disk again, it is best if you reconnect the drive to the power supply and do a cold boot.

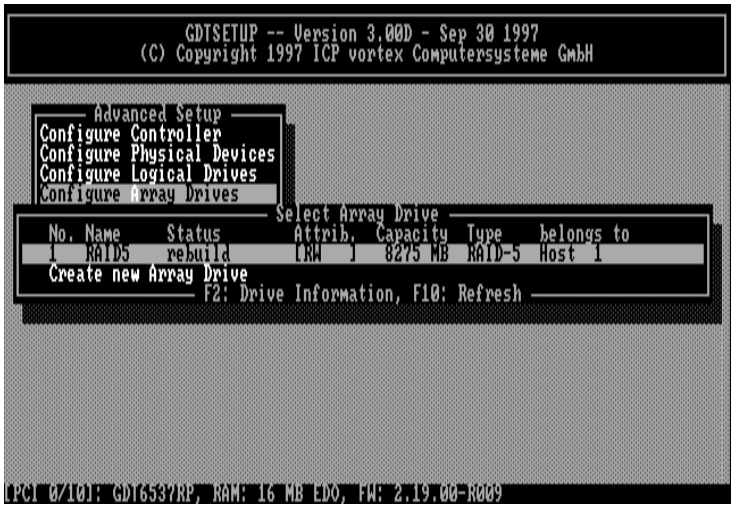
After loading GDTSETUP select the *Configure Array Drives* menu. Select the *Replace Array Component* menu



Press <ENTER> . GDTSETUP recognises the previously failed drive again (it was not really defective) and asks if it should be integrated into the disk array again.



Answer <Yes> and the disk array changes its state into *rebuild*. After leaving GDTSETUP the controller begins the reconstruction of the data of the failed drive.



After the completion of this process, the disk array's state changes into *ready* again.

A few words on the replacement of a defective hard disk of a disk array.
If a hard disk belonging to a disk array for which no Hot Fix drive had been assigned should fail, you should replace this defective hard disk with a new one *as soon as possible*. Always be aware of the fact that this disk array does not have any redundancy until the defective hard disk has been substituted. This means that if another hard disk should fail while the disk array is without redundancy, all data is irretrievably lost. RAIDYNE offers two possibilities of replacing a failed drive of an array for which no Hot Fix drive has been designated:

1. Replacement with GDTSETUP (we have just demonstrated this method)
2. Replacement by using the **Hot Plug function of GDTMON**

The Hot Plug method allows you to replace a defective drive while the disk array continues to work and without having to load GDTSETUP. When this method is used, the GDT SCSI bus to which the defective drive is connected, is temporarily halted (that is, for the time necessary for replacement), thus enabling you to disconnect the defective drive from the SCSI bus without any risk. After the replacement, the SCSI channel halt is lifted again and RAIDYNE automatically begins to rebuild the new drive. The halting and release of the GDT SCSI channel is controlled by GDTMON, which is available for most operating systems supported by the ICP Controller. The above mentioned halt of the SCSI bus serves to avoid that interferences (spikes and glitches) which inevitably occur on the SCSI bus when the defective drive is disconnected, impair the functioning of still intact drives on this SCSI channel. However, this implies that none of the SCSI devices of the halted SCSI channel can be accessed during the time the defective drive is being exchanged. If all drives forming the disk array are connected to one single SCSI channel the entire disk array cannot be accessed during the time of replacement. Therefore, it is evident that the ICP Controller should have as many SCSI channels as possible and that all SCSI devices should be distributed equally to the available channels in order to avoid that the disk array or other SCSI devices cannot

be accessed during the Hot Plug drive replacement. The Hot Plug should be carried out as quickly as possible.
The ICP Controllers also support the Intelligent Fault Bus (DEC ™ fault bus) and SAF-TE. Both industry standards allow for a replacement without any program interaction. Simply unplug the defective drive and plug in a new one. To make use of this very comfortable methods, an intelligent subsystem (supporting either the Intelligent Fault Bus, or SAF-TE) is required.

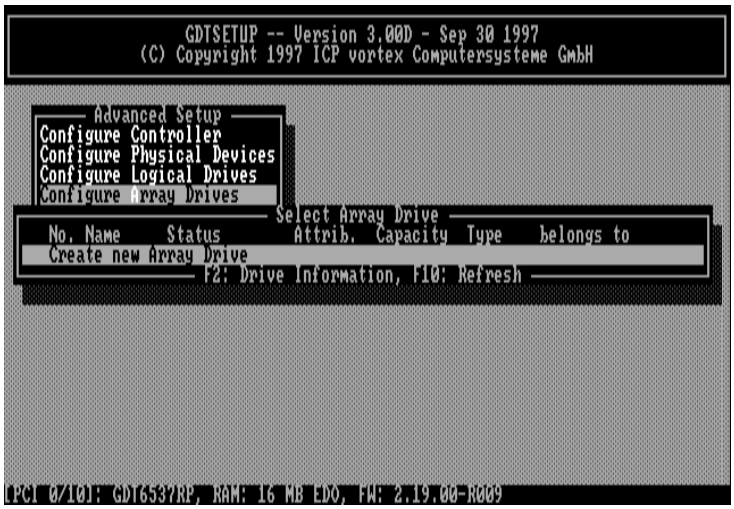
We would like to stress that the Hot Fix method is by far the most secure method of replacing a defective drive while the disk array is operational (see next example). First of all, because it is completely automatic, and secondly because it does not imply any mechanical or electrical interventions on the disk array as the Hot Plug method does.
We shall explain GDTMON and the Hot Plug method more thoroughly later in this manual.

C.9 Example 4 - RAID 5 Disk Arrays With a Hot Fix Drive

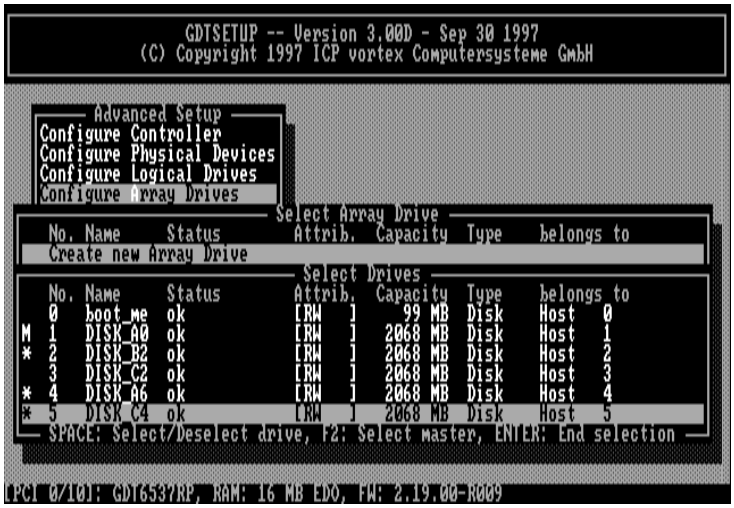
This example is applicable to ICP Controllers with the RAIDYNE firmware. What we call **Hot Fix** drives is referred to as **Host Spare** drives in some literature. Most part of the installation is carried out as in our third example, so we do not repeat the explanation.
Do Step 1, Step 2 and Step 3 as described in example no. 3.

Step 4: Configure Array Drives

We now leave level 2 (by pressing the <ESC>-key) and are back in the main menu. Now, with the cursor keys ↑ and ↓ select *Configure Array Drives* and go to level 3 by pressing <ENTER>.

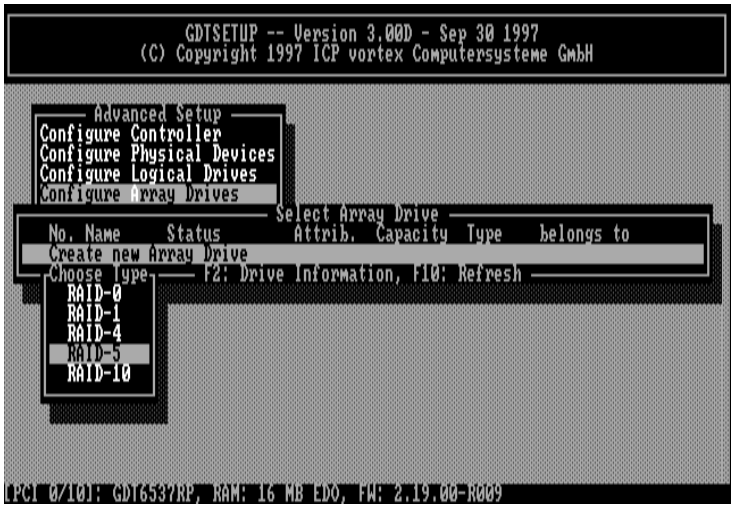


Press <ENTER>. (Note: the Host Drive boot_me is not relevant for our example). Select one by one the Logical Drives 1, 2, 4 and 5. Omit Logical Drive 3.

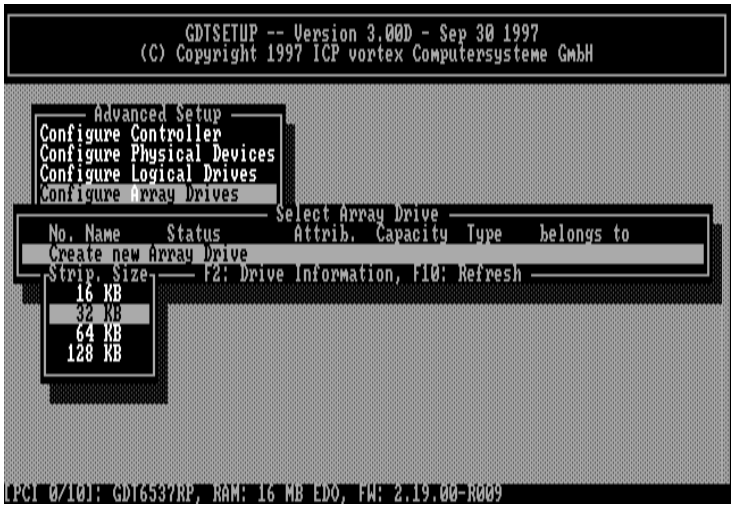


When the last Logical Drive is selected, confirm with <ENTER>. GDTSETUP now displays a list of possible RAID levels available with the number of Logical Drives selected.

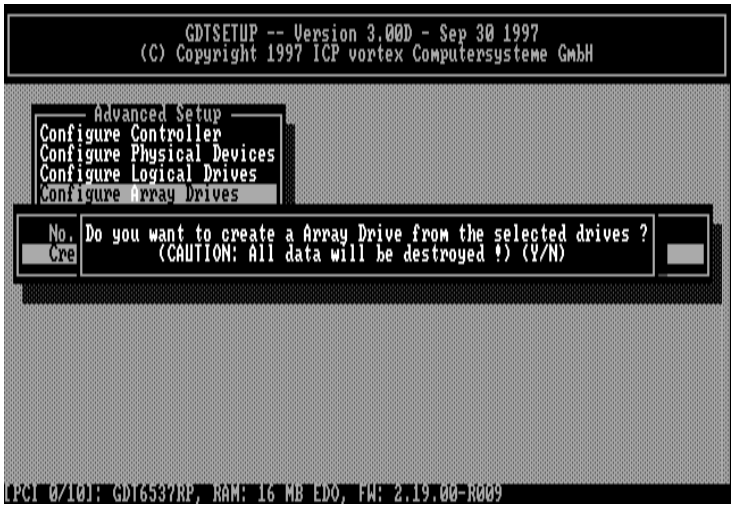
- RAID 0 pure data striping without redundancy
- RAID 1 disk mirroring
- RAID 4 data striping with dedicated parity drive
- RAID 5 data striping with striped parity
- RAID 10 RAID 0 combined with RAID 1



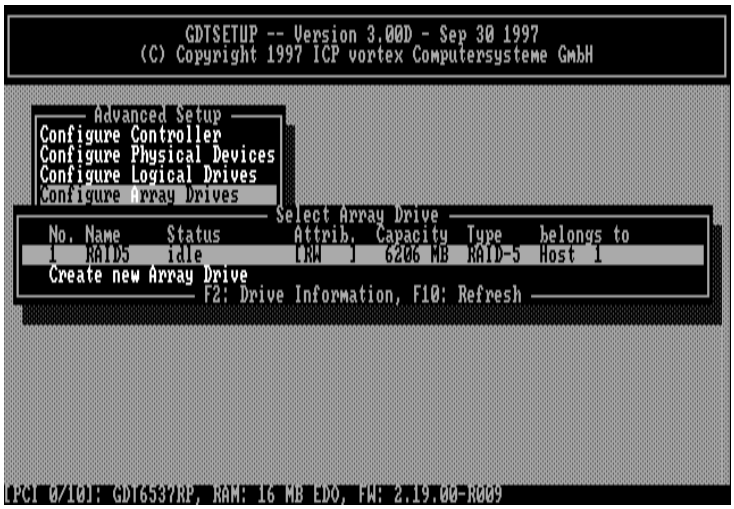
In our case we take RAID-5 and press <ENTER>.



GDTSETUP asks for the *Stripe Size*. This is the size of the stripes into which the data is divided. The default is 32KB which we leave for this example and therefore press <ENTER>. (Note: 32KB stripe size is suggested because in various performance tests it has proved to be the best value.). GDTSETUP displays a security request, which we confirm with <Y>.



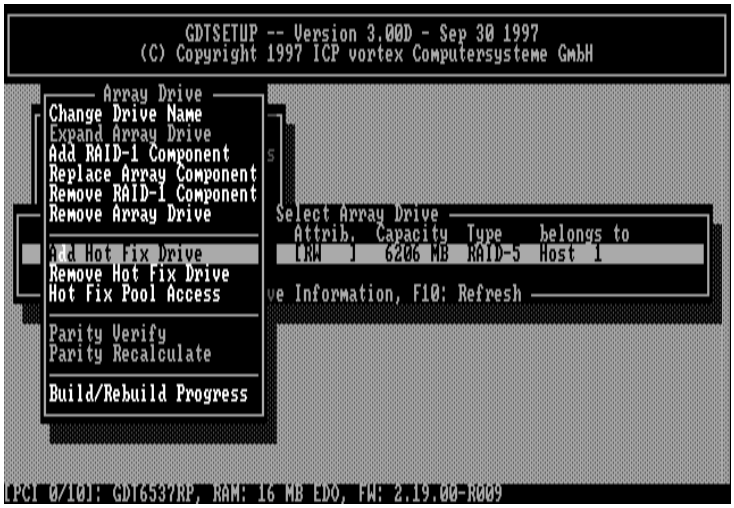
GDTSETUP allows you to limit the capacity of the disk array. This may be of interest if your installation requires an exact size for a disk array. Normally, the full capacity is used. In our example we press <ENTER> .



We succeeded in setting up a RAID 5 disk array. The screen shows that the disk array is currently in an *idle* state. Later in this chapter, we shall explain the different states a RAIDYNE disk array can assume.

Step 5: Creating a Hot Fix Drive

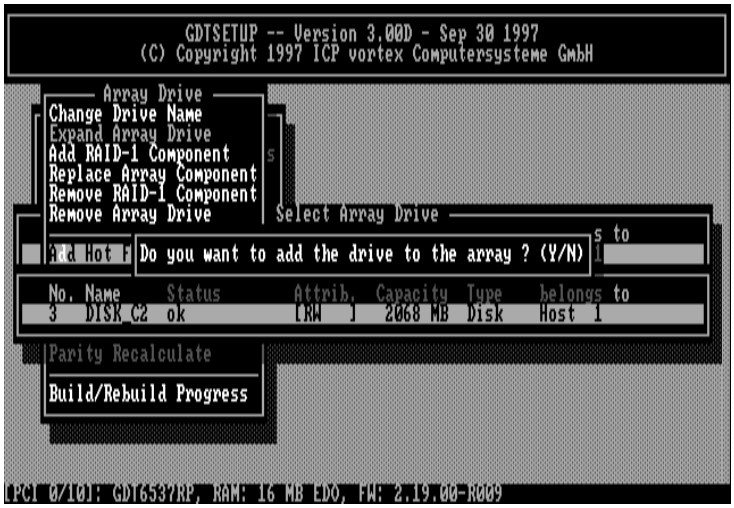
Press again <ENTER> and move the selection bar to the *Add Hot Fix Drive* menu.



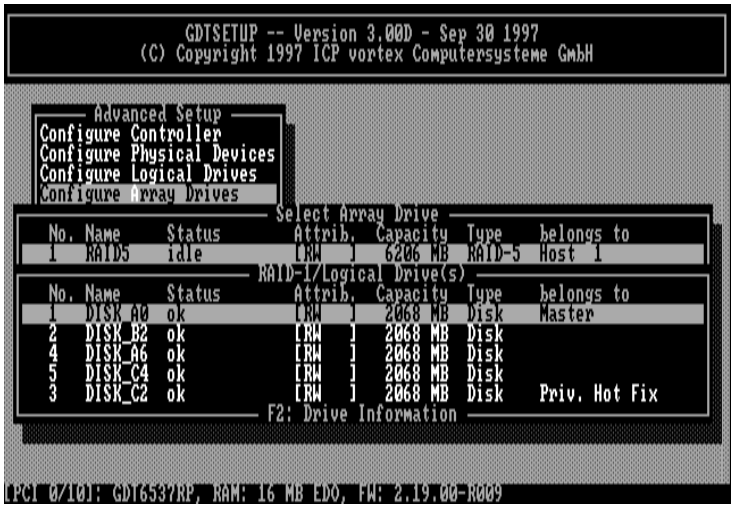
Press <ENTER>. GDTSETUP now displays a new dialog-box containing all the Logical Drives apt to serve as a Hot Fix drive (one criterion for this suitability is the drive's capacity, i.e., it has to be large enough). So do not be surprised if you do not find all the drives you would have expected during later installations. GDTSETUP knows which drives are suited to be used as Hot Fix drives. In our example, GDTSETUP offers the Logical Drive we have omitted during the configuration of the Array Drive.

Press <ENTER>.

GDTSETUP offers two different Hot Fix types: A *private* Hot Fix drive is only available for one specific disk array. A Hot Fix drive in a Hot Fix *Pool* can be made available to several disk arrays (presuming that the capacity fits). In our example we choose the *Private Hot Fix* drive and press <ENTER> .



Attention: By turning a Logical Drive into a Hot Fix drive, all its data is irretrievably lost.



After pressing <F2> we get detailed information on the structure of the disk array. The last entry refers to the *Priv. Hot Fix* drive.
We have already seen this form before, with the only difference that DISK_C2 has been assigned to be the Hot Fix drive. We now leave GDTSETUP as described in example no. 3, in order to allow GDTSETUP to send all relevant information to the controller and let RAIDYNE create and store the redundant information.

The question that arises now is:

When and how does the Hot Fix mechanism work ?
Normally, RAIDYNE puts Hot Fix drives in a stand-by mode, that is, their motors are stopped. However, it may happen that certain operations such as loading drivers, starting GDTSETUP and so on, cause the Hot Fix drives to start their motors. This takes a little bit longer, but it is necessary in order to enable RAIDYNE to check the consistency of the setup. RAIDYNE substitutes a failed hard disk with a Hot Fix drive only if the array was in a *ready* state before the failure. Or, in other words, a Hot Fix drive can only be activated if the corresponding disk array was in a state of data redundancy at the moment of failure.
Important: The following partial step can only be performed if the disk array is in the *ready* state.

Step 6: Simulating a Hard Disk Failure When a Hot Fix Drive is Present.

This partial step is optional. However, we recommend that you carry it out in order for you to get an idea of how RAIDYNE reacts in such a situation and which steps need to be taken. To have a very realistic simulation, create a DOS-partition and generate load on your disk array by using a batch file with copy commands. During these copy operations we cause DISK_B2 to fail by plugging out its power supply. (If you choose to let another drive fail, keep in mind the section of step 1 referring to the SCSI-bus termination).
We now observe how RAIDYNE reacts:

1. After a short while, GDT's alarm signal is heard.
   (Note: the alarm only goes on when the RAID 5 Array Drive is accessed.)

2. RAIDYNE activates the so-called *fail* operation mode. In this mode, the disk array remains fully operational. The data of the failed drive is reconstructed by means of the redundancy information stored on the other drives.

3. RAIDYNE starts the motor of the Hot Fix drive.

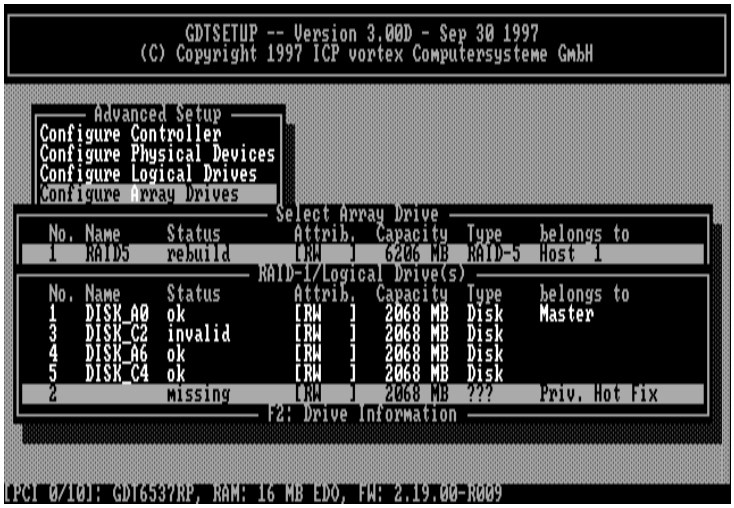
4. RAIDYNE includes the Hot Fix drive into the disk array and starts to reconstruct the data and redundancy information. The disk array is now in the operation mode *rebuild*.

5. The alarm signal is not turned off until a new Hot Fix drive is added to the disk array, or until GDTSETUP (or GDTMON) is loaded and the missing Hot Fix drive is removed or replaced with a new one.

Obviously, no other hard disk may fail until all data is entirely reconstructed on the Hot Fix drive, because up to that moment the system operates without redundancy.

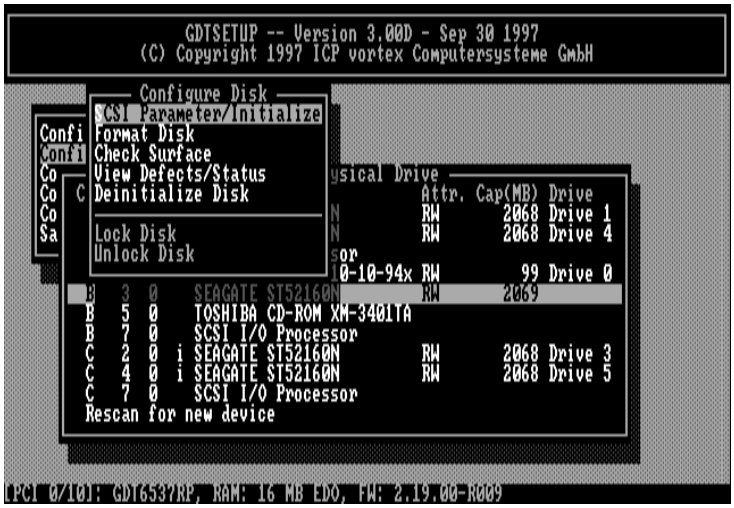
How is this situation reflected in GDTSETUP?
What has happened to the failed drive ?

To answer these and other questions we load GDTSETUP and check. We go directly to the menu *Configure Array Drive* menu. As expected, the disk array is in the *rebuild* state. Request the drive information regarding the RAID5 disk array with <F2>.

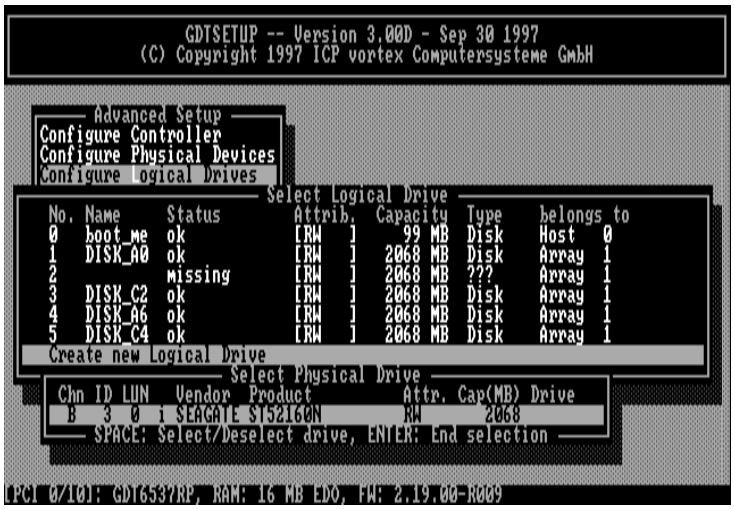


Two changes have occurred. The Hot Fix drive DISK_C2 has the state *invalid* and has jumped into the position of DISK_B2. DISK_B2 is missing (this is the drive we let fail (disconnected it from the DC power). Since this disk array was configured to have a hot fix drive, this position is still in the list, with the attribute *missing*.
Leave GDTSETUP and when you are back at the MS-DOS prompt, switch off the system. We now want to add a new Hot Fix drive to the disk array. For our example we take a brand new

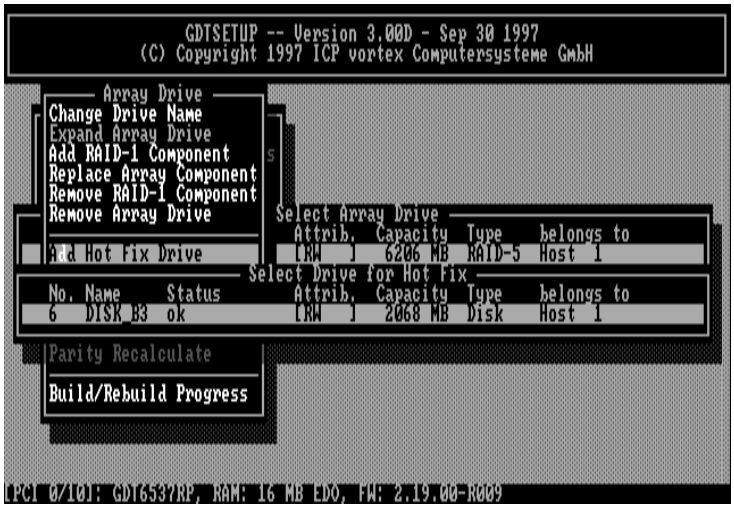
drive, set its SCSI ID to 3 and connect it to the connector of SCSI channel B, where the previously failed drive was connected to and to the DC-power supply. Before switching on the system again, check that the SCSI termination of the new drive is identical to the old one. After switching on the system again, load GDTX000 and GDTSETUP and initialize the new drive.



In the next step we setup a new Logical Drive:



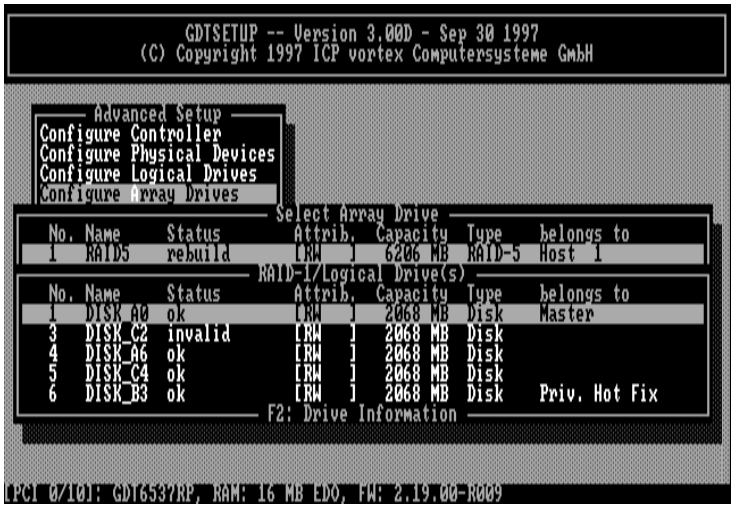
DISK_B3 is our new Logical Drive, which we want to use as a new Hot Fix drive.
Change to the *Configure Array Drives* menu. Select our Disk Array and press <ENTER>.



Select *Add Hot Fix Drive* and thereafter DISK_B3.

Select *Private Hot Fix* and press <ENTER> .
Press <F2> to get the configuration information on this disk array.

As we can see from the list, DISK_B3 has become the new Hot Fix drive for our RAID5 disk array.



DISK_C2 is still *invalid* (this was our former Hot Fix drive), since the rebuild process is not yet completed.

C.10 Trying to Answer The Initial Questions

Now, after having demonstrated with examples 3 and 4 how RAID disk arrays are created with RAIDYNE (we hope you enjoyed it), we would like to return to the questions set down at the beginning of this chapter. When planning a disk array it is essential that you have precise ideas on how you intend to configure the disk array.

C.10.1 How Many Hard Disks Should be Integrated Into the Disk Array ?

To answer this question let us have a look at the delimiting parameters, that is, the maximum and minimum number of drives. The **maximum number** of physical drives in a disk array is determined by the number of physical drives the ICP Controller can control (with a two-channel ICP Controller this number amounts to 14, with a three channel ICP Controller to 21). In this context, we cannot analyze the many various factors which influence the decision of whether to integrate all Host Drives into one single RAID Host Drive, or rather create a number of smaller RAID Host Drives instead. The **minimum number** of necessary hard disks depends on the RAID level you wish to realize.

RAID Level	Type of Disk Array	Minimum number of hard disks
RAID 0	data striping	2
RAID 1	disk mirroring	2
RAID 4	data striping with parity drive	3
RAID 5	data striping with striped parity	3
RAID 10	data striping and mirroring	4

The desired usable disk space of the disk array as well as the following two issues have a direct impact on the number of physical hard disks needed.

C.10.2 Which Level of Redundancy is Needed ?

To come straight to the point, **RAID 0** (data striping) does not imply any redundancy at all (the **R** in front of the **AID** is rather misleading). On the other hand, a RAID 0 disk array is pretty fast, since no parity information is required. With **RAID 1** (disk mirroring), the data is 100% redundant because it is mirrored. This is definitely the highest level of redundancy, but the most expensive one, too. An interesting combination of RAID levels 0 and 1 is **RAID 10.** Two RAID 0 stripe sets are simply mirrored. If one drive fails, the data are still available on the mirrored drive. With **RAID 4** (data striping with dedicated drive) and **RAID 5** (data striping with striped parity), parity information is calculated from the present data with a simple mathematical operation (eXclusive OR, XOR), and stored either to one dedicated drive (RAID 4) or to all drives (RAID 5). If one drive should fail, the data of the defective drive can be reconstructed on the basis of the normal user data and the previously calculated parity data. RAID levels 4, 5 and 10 can tolerate the failure of one drive just as RAID 1, but in comparison to the latter, RAID 4, RAID 5 or RAID 10 are less expensive.
As already mentioned before, the entire disk array controlling function is carried out at controller level and therefore does not load the host computer.
Let us have a look at the following table which explains the correlations between the RAID level, usable disk capacity and number of physical hard disks. To make things easier, we consider identical 1 GB hard disks:

Usable storage capacity of the disk array

RAID Level	2 hard disks	3 hard disks	4 hard disks	5 hard disks
RAID 0	2GB	3GB	4GB	5GB
RAID 1	1GB	1GB	1GB	1GB
RAID 4	-	2GB	3GB	4GB
RAID 5	-	2GB	3GB	4GB
RAID 10	-	-	2GB	-

It is quite obvious that the redundancy of level RAID 1 soon becomes very expensive when more than 2 hard disks are used. Only with RAID 4 and RAID 5 have you a reasonable relation between storage capacity and expenses for the disk array.

C.10.3 Do we Need Hot Fix drives ?

In other words: Should RAIDYNE automatically reconstruct the lost data after a hard disk failure ?
One of the reasons that have led you to choose **R**AID disk arrays definitely lies with the **r**edundancy, that is, the data security you still preserve even in the event of disk failure, thus resting assured against loss of data and time.

Hot Fix drives are possible with all RAID 1, 4, 5 and 10 disk arrays.
In order to assist the following considerations, we define the term ***time without redundancy*, TWR**. Set apart the time needed to set up the disk array (state *build*), the time without redundancy should be kept as short as possible. Let us assume that one of the hard disks of the RAID 5 disk array we set up with example 1 fails. The disk array is without redundancy. TWR starts to run. Any superfluous prolongation of the TWR (because you have to get a replacement drive, or because you did not realize the failure immediately since you didn't hear the ICP Controller's alarm signal, or because nobody checked the file server) increases the risk of data loss which will occur if a second drive should fail. Therefore, new redundancy should be created as soon as possible and in an entirely automated manner. Integrating a Hot Fix drive as an immediately available and auto-replacing drive is the only way to keep the TWR as short as possible. Only a Hot Fix drive can ensure optimal disk array security and constant data availability. Of course a Hot Fix drive is not compulsory. If you control the disk array at regular intervals and immediately replace a defective drive (by shutting down the system or hot-plugging), you can do without a Hot Fix drive.

C.11 States of a RAIDYNE Disk Array

An Array Drive under the RAIDYNE operating system can assume seven different operational modes: ***Idle, Ready, Fail, Build, Rebuild, Expand and Error.***

C.11.1 "Idle" State

This state is characterized by the fact that the redundant information of the disk array has never been entirely created. The disk array is in this state after its first configuration and until you quit GDTSETUP. If an error should occur while the array is in the *build* state, the array returns to the *idle* state (exception: if during *build* mode the dedicated drive of RAID 4 fails, the mode changes to *fail*).

C.11.2 "Build" State

After the disk array has been configured for the first time, it assumes the *build* state as soon as you quit GDTSETUP. While the array is in the *build* state, redundancy information is calculated and stored to the hard disks of the array.

C.11.3 "Ready" State

The disk array is fully operational when in the *ready* state. All redundant information is present, that is, a hard disk can fail without impairing the functionality of the disk array. This is the normal state of a disk array. The state *ready/expand* indicates, that the RAID level and/or capacity are currently migrated/expanded.

C.11.4 "Fail" State

The disk array changes to the *fail* state whenever a Logical Drive fails. Redundancy information is still present, thus allowing the remaining hard disks to continue working. This state should be eliminated as soon as possible by replacing the defective hard disk. If a so-called Hot Fix drive has previously been assigned to a disk array with GDTSETUP, the controller will automatically replace the defective drive and start the reconstruction of the data and the redundant information. Therefore, under these circumstances the *fail* state is only temporary and will be eliminated by the controller itself.

C.11.5 "Rebuild" State

The disk array will assume this state after the automatic activation of a Hot Fix drive or after a manual replacement carried out with GDTSETUP. The data and the redundant information are reconstructed and stored to the new drive.

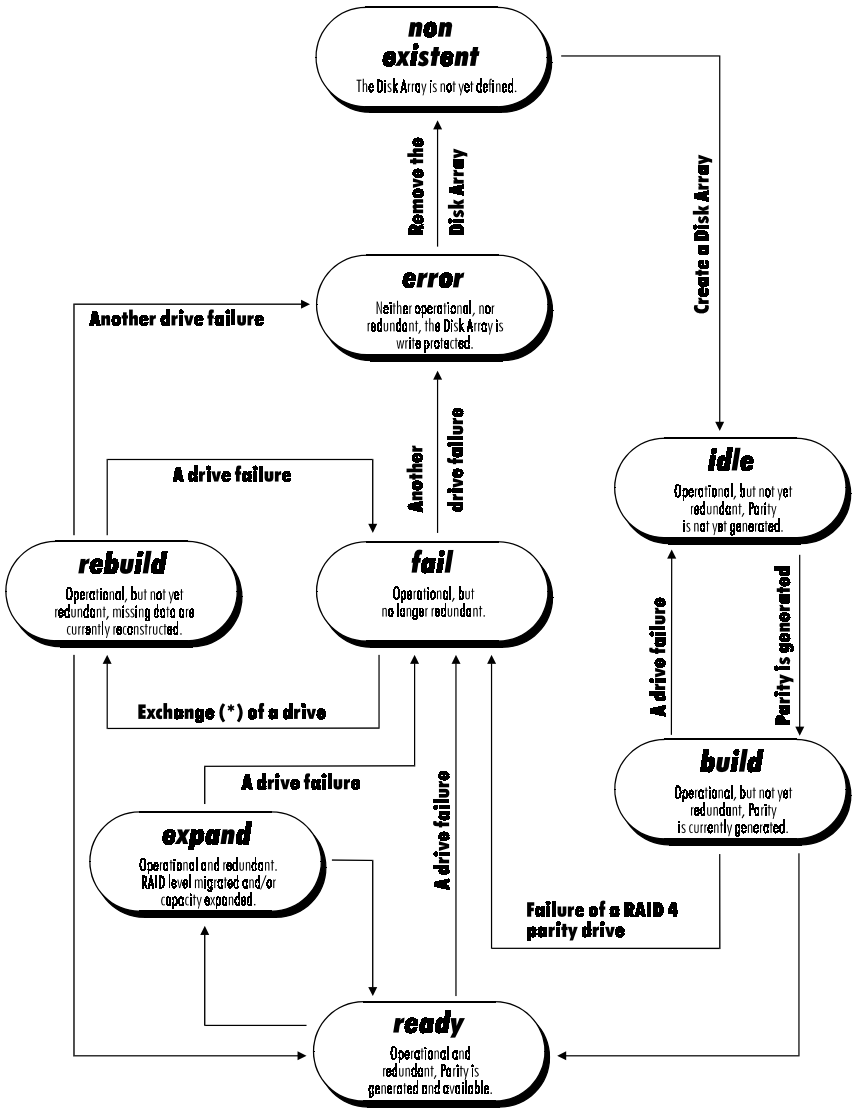
C.11.6 "Expand" State

If the capacity or RAID level of an existing disk array is changed, the disk array changes its state into *expand*. As soon as the expansion or migration is completed, the state changes back to *ready*.

C.11.7 "Error" State

If a second hard disk should fail while the disk array is in the *fail* or *rebuild* state, it is not possible to continue the working session without restrictions. The disk array is still available for I/Os, but data loss and error messages on the host level are possible.
The following state diagram of the disk array summarizes the states described above and the transitions from one state to another.

Some of these states may become the addendum ***patch*** (e.g. *build/patch*, *ready/patch*).
This word indicates that the original Array Drive went through a significant procedure. I.e., the parity information was recalculated anew.
Or, the Array Drive has been patched from the error state into the fail state. This may become extremely helpful in a situation where two Logical Drives of an Array Drive, fail at the same time, but only one of the two Logical Drives is really defective and the other was blocked out, since it was connected with the same SCSI channel as the defective one. The Array Drive's state is error and normally all data would be lost. RAIDYNE and GDTSETUP include some functions, which allow the patch of this Array Drive from the error state into the fail sate. Before the actual patch, the defective drive has to be physically removed from the Array Drive. Such a patch-procedure is a real sheet-anchor and should only be used, after a detailed consultation with a trained support person (a printout of the *Save Information* file, is extremely helpful).



(*) Replacement either manually, or through hot fix method.