System Disaster Recovery: Tips and Techniques

Jason M. Goertz
Systems Engineer
Hewlett-Packard
Bellevue, Washington

INTRODUCTION

Since its introduction in 1973, the HP3000 has proved to be one of the most reliable computer systems ever built. Hardware reliability is extremely good, with a minimum of downtime in the case of most users. An extensive field operation exists in the Customer Engineering Organization which, in most cases, can diagnose and repair failing hardware in a very short amount of time. For the really sticky problem that does occur, there exists a large team of engineers in the various manufacturing divisions which backs up the field personnel. This, coupled with a computerized parts inventory system in the field, assures the user that the fastest possible repair will be made to his system. Software reliability is also very good. This is due in a large part to HP's policy that the operating system and subsystems will not be modified by field personnel or by customers, as is the case with many vendors. Along with this, software distribution is handled by the local Field Software Coordinator, giving the field a fair amount of leeway in exactly what software is released to an individual area, while maintaining a reasonable amount of central control.

Unfortunately, all things created by the hand of man are built with imperfections, and this includes computer systems. This paper will deal with the event that these imperfections manifest themselves in such a way as to destroy, or threaten to destroy, the integrity of the system and, of more importance, the data stored on the computer system. This is always more valuable to the owner of the computer than the machine hardware itself.

Causes of this system (and data) integrity loss are many and varied. Usually, a severe hardware failure such as a disc head crash, will cause data to be lost. Many times, a natural occurence (an "Act of God" as the service contract puts it) against which the hardware cannot protect itself will be the culprit. An example might be a severe lightning storm which causes power fluctuations or surges. Software is not free from blame either. Software failures are generally caused by a specific bug which has not been fixed, usually in the operating system. A recent example of this is the PTAPE intrinsic. There were calls to ATTACHIO in PTAPE that were hardcoded to write data to areas in virtual memory on LDEV 1. With the advent of multi-spindle virtual memory on MPE-IV, this became a disasterous situation. A data segment could be built on LDEV 2 or another system domain disc, but PTAPE would write it to the corresponding location on LDEV 1, causing a clobbered directory, system code etc. Many times human beings are the cause of integrity problems. A good example of this is stopping the system while it is in the process of coming up, for whatever reason. This can, and usually does, result in a system which will not boot at all.

In any case, loss of integrity results when the system cannot be started. In other words, INITIAL will not complete the startup procedure, and the user is left with a system that will not come up, with all his data on disc and apparently inaccessible. Or is it? This paper will attempt to describe how to recover the system, and the data.

In the following pages, we will discuss the following topics:

1. How to prepare for a system disaster.
2. Reasons for loss of system integrity.
3. What to do to recover data.
4. Description of utilities to help prevent system problems, and recover the system should this become necessary.

It should be remembered that many of the suggestions presented in the following pages are the "ideal case," and are not absolutely necessary for a well managed system. Many users, for reasons of economy or time, cannot follow all of these suggestions to the letter, and do quite well with their systems.

PREPARATION FOR DISASTER

The following are suggestions for what to do before the system gets into a state where it is unusable. While some of them may seem quite obvious, it is surprising how the basics are often times overlooked. As with almost anything, "be prepared" is the rule of the day if one is anticipating a bad situation.

In all but a few cases, a system which will not start probably has corrupt system files and data. In a few

specific instances (covered later) this can be fixed and the system brought up. However, 95% of the time, a reload is in order. It is important to realize this fact. A reload may, and probably will, be necessary to recover the system. This point cannot be emphasized enough. The only good way to build a corrupt system is not to try to fix what is wrong, but to totally start over and build a new system from scratch. Many people try to avoid this step and find themselves in an even worse situation than before. It is very important that the system manager who finds himself having to recover the system accept the fact that the system will be down for a while.

Doing a reload may not be as bad as it sounds. There are five options on a reload: SPREAD, RESTORE, COMPACT, ACCOUNTS, AND NULL. A full description of what these do can be found in section 6 of the System Manager manual. Briefly, though, SPREAD, RESTORE, and COMPACT will attempt to load all files onto the system. Since the typical system manager will be doing the recovery in the midst of angry users ringing the phone off the hook, it is to everyone's best interest to get the system up as soon as possible. Thus, the ACCOUNTS or NULL option, which do not load all files, should be used. A later section will deal with exactly when to use which of these two options.

In order for a reload to be done, however, there must be something to reload from. This brings us to the first and by far the most important preparation to be performed. That is, have a good Sysdump set available at all times. Again, this point cannot be overemphasized. We will define a "good Sysdump" as being one that has an intact and up to date version of MPE on it, a good directory, and the most recent user files on it.

The most important contribution toward this goal is to perform regular Sysdumps. A periodic schedule of dumps is highly recommended, and must be adhered to. A full Sysdump every day would be ideal, but many users simply cannot afford this, in terms of system down time, operator cost and tape cost. The next best solution to this is to do a weekly Sysdump, then partials to the last full on the other days of the week. Generally, most users do the full dumps on Monday, Friday, or on the weekend. This is the most common method, and provides adequate Sysdump coverage at a minimum of downtime and cost.

There is one additional thing the Sysdump provides which many users overlook, and that is the listing that comes out when the dump is over. This listing includes the file dumped, what disc it was on, what the disc address was, and what reel of the dump it was stored on. Appendix A contains a sample output. As we will see later, this document is essential to being able to recover data. Many times, people do not generate the list because it is too bulky, etc. It is worth the trouble! It's one of those things you will desperately need when you can't get it. It is handy, too, for finding which reel of the dump a particular file is on when it is necessary to restore one during normal system operation. Store all listings for any Sysdump set that is currently valid. It is a good idea to keep the full dump listings together with other full listings, partial listings with partials.

It is very important to have the Sysdump tapes, as well as the Sysdump listings, stored in a location where they are readily accessable to the person recovering the system. Many users, just for safety, store the previous full set offsite. This is a good idea, since a fire could wipe out all hardware and tapes. An offsite copy would insure that some kind of system could be rebuilt, even if it were a week old.

The Sysdump is of little use, however, if the the data on it is unreadable. Use newer tapes if possible for the backups. At least keep the tapes cleaned regularly, and don't use a tape more than a few times before being cycled out of the Sysdump sets. This, of course, varies with how often the tape is used and the quality of the tape. There are a few programs in existence to verify that a Sysdump tape set is good, which will be described in detail later under UTILITIES. These, at best, only give an idea that the tape is good, since only the file labels are really checked for integrity. But at least parity errors will usually be detected.

In addition to the Sysdump tape sets, the system cold load tape should be kept onsite at all times. This tape is a special form of the Sysdump tape, usually containing only MPE, system and subsystem files (@.PUB.SYS). This tape is generally made by the account SE at the time a new version of MPE is loaded onto the system. It is advisable to keep the cold load tape for the current version of MPE, as well as the previous version, just in case it becomes necessary to go back one release. The reason this tape is valuable is so that if problems arise, a known, "good" version of MPE can be loaded onto the system. This can be done with the UPDATE option. Most users keep an additional cold load tape, this one reflecting all configuration changes. This is a good idea, although the UPDATE from the HP made cold load tape will not affect the configuration. If this additional tape is desirable, an UPDATE from the HP made tape is advisable before the configuration changes are made. This is to prevent any "glitches" in the version of MPE on disc to propogate through the cold load tape(s). Along the same line, an update before each full sysdump is also a good idea, for the same reasons. This has an additional benifit, that being that regular cold loads are performed. My Customer Engineer friends tell me that the customers that do a cold load regularly have fewer problems than users who don't.

An absolutely necessary tape set to have is the diagnostic tape set. For Series II/III, this consists of two tapes, a CPU diagnostic tape, and a Non-CPU diagnostic tape. On the HP-IB machines (Series 30/33/44/64), one tape or floppy is used, called the Diagnostic Utility system (DUS). The DUS contains both CPU and peripheral diagnostics, including SADUTIL and SLEUTH, combined. CPU diagnostics are used almost

exclusively by the CE's, while the Non-CPU diagnostics can be utilized by users. Indeed, this paper deals primarily with that very subject. Again, this diagnostic tape or tapes must be created when the system is operable. Waiting till "later" to make the tape could spell trouble. Whenever a new version of MPE is installed, a new set of tapes (or DUS) should be created. This is because the diagnostics are updated along with MPE and the subsystems.

Finally, it is a good idea to keep some sort of list of accounts, groups, etc that the system currently has. A :REPORT listing would serve very nicely. While this would not be necessary to recover the system, it will serve as reference from which to decide which accounts to reload first. This would be the case if production accounts were to be brought back online before development. Some suggestions as to what to put on this list would be:

1. Prioritization of accounts, groups etc that are to be recovered, so that the most critical can be brought back first.
2. A list of all critical files that might have to be recovered. This should be a list of MPE file names, so databases should be listed as DB01, DB02, etc.

REASONS FOR LOSS OF INTEGRITY

We have seen the types of things that are necessary to prepare for a system integrity loss. We will now discuss exactly what causes the system to become inoperable, and what we can do to bring the system back, and recover all the data.

As has been said before, the time when danger of system integrity loss is highest is when the system is down and will not come up. While it is possible to have a running system and have most of the data corrupt, at least MPE is still running, and we have the aid of it and a host of utilities, plus the ability to restore older versions of files, etc. Most sites do not routinely bring the system down at night. Instead, they leave all hardware powered on and MPE running. The question is really one of how the system gets to the down state. The most common cause here is the system interruption, or more specifically, the system failure, system hang, and system halt.

A system failure occures when some part of the MPE system calls a procedure called SUDDENDEATH. An integer number is passed to SUDDENDEATH, which is printed in a system failure message along with the current hardware status and return address of the calling entity. These calls are placed in the code by the MPE lab purposely, and are used when an "impossible" situation is encountered, and MPE cannot continue running. The system hang can be caused for an infinite number of reasons. It usually ends up being caused by a hardware resource which ceases to function. Eventually, every user on the system asks for that resource, causing everyone to wait. A slightly different variation is when a hardware device ties up a system table, and everyone suspends when they try to access that table. A silent halt is similar to a system hang, except that the hardware is in a state in which it cannot run. During a system hang, the hardware will run, but since everyone is suspended, it is never asked to. Silent halts are generally caused by bad hardware, although there are a few software problems that can cause them.

The standard way to recover from any of the above three system problems is to:

1. Take a memory dump.
2. WARMSTART.
3. Print off all spoolfiles.
4. Load system with UPDATE option from a good cold load tape.
5. Log failure and recovery action.

Step four is one reason why it is a good idea to keep a good cold load tape available. The cold load will get the system back to a known good copy of MPE, which may remove the source of the problem. If the failure then re-occurs, a more serious problem is indicated, and the local PICS center should be consulted. Step 5 is often overlooked in the haste of the moment. It is important to keep track of what kind of failure occured, and what actions were taken in case the problem escalates in severity. The Gold book supplied with the system is a good place to log these facts, and places are provided under "Historical Records." This becomes very valuable to the SE/CE who must try to piece together a history of the system's problems in order to locate any trends. This history is absolutely essential to correcting certain very sticky system problems.

One thing to be noted here is that the failure (hang, halt) is not necessarily the cause for the system integrity loss. In other words, the failure itself does not go out and cause data to be corrupted. What the failure can do is indicate the source of the problem. For instance, something might have, at some point in time, caused the file label of a system file to be destroyed. When the system tried to access that system file, a system failure occured. If the file is a critical one, such as an IO driver, it is probable that the system would not come up. The failure did not cause the system file corruption, but indicated that the file was, indeed, corrupt. In the process, the system got into a state in which it could not or would not start. A variation of this is when the system interruption occurs during the updating of a critical resource. The net result is the same: a corrupt system. The system is now down and will not come up. In the above example, a cold load may have solved the problem. But let us assume it would not. We now have a system which has a corrupt operating system, and will not come up unless the system is rebuilt from scratch. In other words, we have to do a reload. Thus, system failures can lead to the situtation in which user data must be recovered.

WHAT DO WE DO NOW?

Before discussing how we would save the files, a couple of special "system won't come up" cases should be discussed. One is the situation in which the system was stopped in the middle of a startup. This usually happens when the person doing the startup is in a hurry, and in his haste aborts the startup before it has completed. When another start is attempted, INITIAL displays a message saying something to the effect of: "ALL VOLUMES NOT PRESENT. MOUNT CORRECT VOLUMES OR RELOAD," or "VOLUME TABLE DESTROYED — MUST RELOAD." The reason for this is that in the startup process, INITIAL does several things to insure that the system is in a startable state. It updates the Cold Load ID, which is a number that is stored in the system and changed every time the system is started. The Cold Load ID is kept in many places, and only at the end of the startup procedure can we be sure that all places have been updated with the new number. One of the locations the Cold Load ID is kept is the Volume Table, which has a listing of all disc volume names. If the Cold Load ID in the Volume Table does not match the Cold Load ID kept in the disc volume label, (sector 0 of the disc), then INITIAL assumes something is awry, and will not let the system come up.

In most cases, the system can be brought up safely at this point. The problem is to get INITIAL to ignore the Cold Load ID's. This is done by zeroing out the cold load id's, using the HP utility SADUTIL, which will be discussed in detail a little later. Below is a list of the locations of the Cold Load ID that INITIAL checks, plus a few other things that should be set on disc.

1. Word 7 of Sector 0 of every system domain disc.
2. Word %12 of the Disc Cold Load Table, located on LDEV 1.
3. Word 1 of the Volume Table (on disc). The Volume Table is pointed to by words %124-125 of the Cold Load Information Table.
4. Word %32 of the Disc Cold Load Table contains bits which tell what the previous load was. While not absolutely necessary, this should be zeroed out.

A full summary of this procedure is in Appendix B of this paper. It should be noted that this procedure will not always work, and the locations of the data on disc can be changed by the MPE lab at any time. This is, at best, a kludge which can get a system up and running in very few cases.

Another special case is that the directory itself is corrupted in such a way as it needs to be rebuilt. In this case, it is necessary to build an empty directory (which the ACCOUNTS option does not do) and then rebuild the accounting structure. The easiest way is to use a utility called BULDACCT. This creates two jobstreams. The first builds all the accounts, and the second logs on as the manager of all those accounts and builds users, groups, private volumes, etc. The full sequence of events for this would be:

1. Do a full Sysdump. This will have a corrupt directory on the tape, but that doesn't matter.
2. Log on as MANAGER.SYS,PUB and run BULDACCT. This creates two files, JOBACCT and JOBACCTB.
3. :STORE these two files on a separate tape.
4. RELOAD from the full Sysdump, using the NULL option. This builds a system with PUB.SYS, MANAGER.SYS and only system files.
5. :RESTORE the two files off the tape.
6. :STREAM JOBACCT. This will build the accounts, then stream JOBACCTB, which builds the rest of the account structure.
7. :RESTORE @.@.@ from the full sysdump.

Again, this is only a special case. Usually, the system can't be patched together like this, and a reload is in order. It should be noted that BULDACCT is a user written, unsupported utility.

What happens when, after all you try to do, the system still won't come up? How can data be recovered? The main thing that can be done at this point is to reload the system using an ACCOUNTS option, then restore the most critical user files first, and get the most important applications up and running. To do this, use the list of accounts, files, and users that was discussed in Preparation for Disaster. The critical issue is how to recover any data that may have been updated since the last backup. In some cases, there may have been no updating of files, then there is no problem. Another case may be that the transactions lost may be easy enough to recreate that recovery with SADUTIL is not warranted. In either case, bring the system up as quickly as possible.

At this point, data recovery of the system is critical. To take the data off of disc and store it to tape, we use a utility called SADUTIL. This is a standalone utility which is on either the non-CPU diagnostic tape (Series II/III) or the Diagnostic Utility System Tape/Floppy (HP-IB machines). SADUTIL is written so that all the important functions of MPE, such as the ability to talk to IO devices, read and write to disc, and interpret commands are all contained in one program. Indeed, SADUTIL is essentially a small MPE. In addition, it must fit in Bank 0 of memory. For this reason, SADUTIL does have some limitations, which will be discussed later.

SADUTIL has many functions, but the primary ones that we are concerned with are the SAVE function, which takes files off of disc and writes them to tape; the PDSK function, which prints areas of disc; and the EDIT function, which allows disc locations to be modified. SADUTIL, as well as several other very handy MPE utilities, is fully documented in the MPE UTILITIES manual.

We will assume that the diagnostic tape/DUS is made, (which won't be possible if the system is down) the first step is to load SADUTIL. On Series II/III, this is done by the front panel. On the HP-IB machines, the LOAD button is pressed, and the DUS is loaded into memory. SADUTIL is then selected, and is run by the diagnostic loader. SADUTIL then asks if any configuration changes are to be made. All discs, including floppy drives, must be configured at this point. SADUTIL does not look at the MPE configuration files, but rather has its own internal configuration array. This array can be changed later by using the CONF command. After all configuration changes have been made, SADUTIL prompts the user for a command. This could be a SAVE, PDSK, EDIT or any other SADUTIL command. These commands are listed in the MPE Utilities manual SADUTIL section. Some commands require additional dialog, while others do not.

To save files, enter the SAVE command, and SADUTIL prompts for either the file name, or the disc address. Notice that using the file name assumes the directory is intact. It is better to use the disc ldev and address, which can be obtained by the Sysdump listing. If the file is one that has been created since the last Sysdump, then it will be necessary to use the file name, or use the PFIL command to obtain the disc address. Both of these options assume the directory is intact.

One of SADUTIL's limitations should be mentioned, as it can affect the way recovery can be done. First, SADUTIL cannot handle tape switching. This means that if a file set is given which will span more than one reel of tape, the recovery will terminate. The list of important files to recover mentioned in the Preparation section should include a filesize for each file listed. This is so the proper amount of tape can be estimated. It is very important to back large files one at a time, putting them on a separate tape. It is possible to enter the names one at a time to the SAVE command, and only terminate the list when the end of the tape is near. If a file does spill over, then SADUTIL must be restarted. The file will not be damaged on disc, but the copy on tape cannot be used. Therefore, that file must be saved again on another tape. Be sure and keep a written log of what files are saved in what order on what tapes. This is useful later when these tapes are used to restore the files.

After all files have been saved by SADUTIL, the system must be reloaded. As a rule, the ACCOUNTS option is the fastest way to reload and get applications back online. Before starting the reload, there is one thing that should be done to insure that a complete reload is performed. INITIAL will not reload all of MPE (ie, it assumes that some MPE on disc is valid) if the disc volume label is good. Therefore, it is a good idea to force initial to bring all of MPE off of tape by destroying the volume labels on the system volumes. This is done by using SADUTIL's EDIT command, or use SLEUTH(Series II/III) or SLEUTHSM(HP-IB machines). Appendix C has a sample dialog of how this is done with SLEUTHSM. SLEUTHSM is documented in the Diagnostic Manual Set.

After the ACCOUNTS reload is done, the files are :RESTORED back to disc, partial tapes first, then the full tapes. The full tape should be restored with the KEEP option on the :RESTORE, to insure that files do not get written over by older versions of the same files. This is where the prioritized list of files, accounts, groups, etc, comes in handy. Restore the files in order of prioritization, and this will guarantee the shortest time to applications being back online. This list may be deviated from, since how critical an application is can vary drastically. An accounts payable application will not be as critical if bills were payed the day before as it might be otherwise. After the restore(s) are done, the files must be restored from the tape(s) created by SADUTIL. This is done using the utility RECOVER2. RECOVER2 will prompt for file sets and names, and give the option of keeping files already on disc. Always overwrite the version on disc, since the file on the SADUTIL tape will always be more current than the version on the last partial. Appendix D has a full SADUTIL dialog, showing how to list and save files, then how to run RECOVER2 when the system is up.

UTILITIES

As we have seen, in order to recover files off of disc and perform other functions, proper software tools must be used. We have discussed two of these, SLEUTH and SADUTIL. These, however, are of use only when the system is being recovered. It should be emphasized that prevention is more important than cure, and that all bases must be covered before disaster strikes. The following is a list of the Utilities that exist for prevention, and their function. Some are not HP supported, and should be used with the same caveats as any other unsupported utility, such as SOO, IOSTAT, etc.

SLEUTH — Standalone diagnostic that exercises peripheral devices. Used primarily to format disc packs. HP supported.

SADUTIL — Standalone diagnostic that allows file recovery when system is down. Also allows modification of disc areas, disc condensation, printing of file information (variation of :LISTF). HP supported.

RECOVER2 — Used to restore the tape created by SADUTIL. Used after system is up and running. HP supported.

BADLABEL — Checks validity of disc files. Tells if anything is wrong with a file label, including whether or not the extents point to free space, or to the extents of another file. Used as preventative measure. User written, privileged.

VALIDATE — Checks sysdump tape to see if file labels are valid. Checks to see if parity of directory or MPE portion of tape is good. Also prints

out creator data, etc. Similar to old utility, STAN. User written.

BADFILE — Used to tell what the last file on a Sysdump tape is, if Sysdump aborts. User written.

FLUTIL3 — Used to display and modify any portion of a file label on disc. Used also to purge any bad files. User written, privileged.

BULDACCT — Used to rebuild accounting structure of system. Will not always work if directory is corrupt. User written.

GETFILE2 — Used to restore files off of :STORE and Sysdump tapes if creator does not exist on the system. If run with PARM=1, a SADUTIL tape can be restored. User written, privileged.

DISKED2 — Utility which performs the SADUTIL EDIT function. Allows online modification of disc locations. HP supported.

CONCLUSION

In summary, there are several steps which lead to maintaining system integrity, and to recover it if lost:

1. Do consistent, regular backups. Validate tapes to insure that they are readable to the system.
2. Maintain a library of all documents and software necessary to recover the system. This includes full listings from Sysdumps, current diagnostic tapes, and listings of critical application information (file names, etc.)
3. Keep accurate records of all system interruptions, and what action was taken.
4. If the system won't come up, use SADUTIL and SLEUTH to recover files.
5. Reload system using ACCOUNTS option, then restore critical files first. Use RECOVER2 to restore files saved with SADUTIL.

Following these suggestions will, along with some common sense, provide the necessary procedures to insure that the HP3000 provides quality service to the users. My sincere wish is that no one will ever have to use the information in this paper.

APPENDIX A

The following is a sample output listing of Sysdump. This shows the file dumped, where it is located on disc, and what reel of the dump it was stored on.

```
FILE      .GROUP    .ACCOUNT    LDN   ADDRESS   VOLUME

BADLABEL.PUB        .GOERTZ     1     %73067      1
BANNER  .PUB        .GOERTZ     1     %73143      1
COPYLIB .PUB        .GOERTZ     1     %73400      1
COPYLIBK.PUB        .GOERTZ     1     %75754      1
CRASH   .PUB        .GOERTZ     1     %73340      1
CRASH2  .PUB        .GOERTZ     1     %73351      1
CRASH2P .PUB        .GOERTZ     1     %77257      1
CRASHP  .PUB        .GOERTZ     1     %77324      1
CRASHU  .PUB        .GOERTZ     1     %146011     1
DBBUFFER.PUB        .GOERTZ     1     %77371      1
DBWIZARD.PUB        .GOERTZ     1     %77407      1
DECOM3  .PUB        .GOERTZ     1     %146322     1
DECOMP6 .PUB        .GOERTZ     1     %146453     1
DIRMATCH.PUB        .GOERTZ     1     %73223      1
DISCADDR.PUB        .GOERTZ     1     %77507      1
DSCAN   .PUB        .GOERTZ     1     %77522      1
DSCANTST.PUB        .GOERTZ     1     %146604     1
DUANE   .PUB        .GOERTZ     1     %146616     1
DWNTST  .PUB        .GOERTZ     1     %147156     1
DWNTSTP .PUB        .GOERTZ     1     %147175     1
ENTRYPNT.PUB        .GOERTZ     1     %147203     1
EVERGRN .PUB        .GOERTZ     1     %147213     1
EXAMPLE .PUB        .GOERTZ     1     %147275     1
EXAMPLEP.PUB        .GOERTZ     1     %147465     1
FLABEL  .PUB        .GOERTZ     1     %147476     1
FLIMIT  .PUB        .GOERTZ     1     %147642     1
FLUTIL3 .PUB        .GOERTZ     1     %147705     1
FORTRAN .PUB        .GOERTZ     1     %147764     1
```

FTNLIST	.PUB	.GOERTZ	1	%227775	1
FTNNEW	.PUB	.GOERTZ	1	%77614	1
FTNUSL	.PUB	.GOERTZ	1	%232341	1
GETSTRNG	.PUB	.GOERTZ	1	%150564	1
GRDSCHMA	.PUB	.GOERTZ	1	%150572	1
GRDTEST	.PUB	.GOERTZ	1	%154731	1
GRDTSTF	.PUB	.GOERTZ	1	%154740	1
ID	.PUB	.GOERTZ	1	%154755	1
IDGGEN	.PUB	.GOERTZ	1	%154766	1
KSAMRBLD	.PUB	.GOERTZ	1	%232652	1
LABJOB	.PUB	.GOERTZ	1	%154777	1
LIBREST	.PUB	.GOERTZ	1	%232714	1
LIMCHNG	.PUB	.GOERTZ	1	%155006	1
LSTALLOC	.PUB	.GOERTZ	1	%162677	1
PICS	.PUB	.GOERTZ	1	%162707	1
PICS2	.PUB	.GOERTZ	1	%162714	1
PICTEST	.PUB	.GOERTZ	1	%162720	1
PICTEST1	.PUB	.GOERTZ	1	%233025	1
PRINTER	.PUB	.GOERTZ	1	%233040	1
QDISPLAY	.PUB	.GOERTZ	1	%233054	1
RTMX	.PUB	.GOERTZ	1	%233140	1
SEGPROG	.PUB	.GOERTZ	1	%234675	1
SEP329	.PUB	.GOERTZ	1	%234710	1
SETCOBOL	.PUB	.GOERTZ	1	%234716	1
SETTDPC	.PUB	.GOERTZ	1	%234723	1
SL	.PUB	.GOERTZ	1	%234731	1
SL2	.PUB	.GOERTZ	1	%235076	1
SLPMAP	.PUB	.GOERTZ	1	%235243	1
SLPMAPP	.PUB	.GOERTZ	1	%235440	1
SLPMAPQ	.PUB	.GOERTZ	1	%235466	1
SM	.PUB	.GOERTZ	1	%235646	1
SOO	.PUB	.GOERTZ	1	%237704	1
SPD4800	.PUB	.GOERTZ	1	%235663	1
SPL	.PUB	.GOERTZ	1	%240347	1
SPL2	.PUB	.GOERTZ	1	%235671	1
SPLLAB	.PUB	.GOERTZ	1	%240073	1
SPLLAB2	.PUB	.GOERTZ	1	%235715	1
SPLLAB3	.PUB	.GOERTZ	1	%241327	1
SPLSTD	.PUB	.GOERTZ	1	%240117	1
SPLXREF	.PUB	.GOERTZ	1	%241542	1
SUSTRACK	.PUB	.GOERTZ	1	%241620	1
SWITCH	.PUB	.GOERTZ	1	%241650	1
TAPELAB	.PUB	.GOERTZ	1	%241664	1
TERMID	.PUB	.GOERTZ	1	%241676	1
TEST	.PUB	.GOERTZ	1	%241704	1
TESTER	.PUB	.GOERTZ	1	%241711	1
TESTFILE	.PUB	.GOERTZ	1	%241717	1
TESTVM	.PUB	.GOERTZ	1	%244257	1
UDC	.PUB	.GOERTZ	1	%275247	1
UDCCPL	.PUB	.GOERTZ	1	%235731	1
UDCUTIL	.PUB	.GOERTZ	1	%244435	1
ULDSET	.PUB	.GOERTZ	1	%275325	1
USERINIT	.PUB	.GOERTZ	1	%275336	1
UT817	.PUB	.GOERTZ	1	%275350	1
XXX	.PUB	.GOERTZ	1	%275417	1
XXXP	.PUB	.GOERTZ	1	%275601	1
XYZP	.PUB	.GOERTZ	1	%275611	1

APPENDIX B

The following is a sample dialog showing how to use SADUTIL to set the cold load id's on disc. This is in the event that a load was aborted. This dialog was done on a Series 44 with one 7925 disc.

```
->start
}
HP32033C.F0.D3
WHICH OPTION <WARMSTART/COOLSTART>? COO
ANY CHANGES?
STACK MARKER TRACE
     1     3552   100035      174    MAINSEG1
     0       13   101037        4    BOOTSTRAP


ERROR #201  VOLUME TABLE DESTROYED - MUST RELOAD

HALT  4
->LOAD
%"%

Diagnostic/Utility System Revision 01.01
Enter Your Program Name (type HELP for program information)
:SADUTIL
Disc Utility A01.03 (C) Hewlett-Packard Co., 1976
LIST LOGICAL DEVICES? Y
 LDEV   DRT   UNIT   TYPE   SUBTYPE
 ----   ---   ----   ----   -------

DISC CONFIGURATION CHANGES? Y
LOGICAL DEVICE? 1
DRT? 89
UNIT? 0
TYPE? 0
SUBTYPE? 9
LOGICAL DEVICE?
LIST LOGICAL DEVICES? Y
 LDEV   DRT   UNIT   TYPE   SUBTYPE
 ----   ---   ----   ----   -------

   1     89     0      0        9
SERIAL DEVICE CHANGES? Y
DRT? 73
UNIT? 0
TYPE? 24
SUBTYPE? 0
ENTER FUNCTION: PDSK 1
 ENTER ADDRESS: 0,1;A
 LDEV=    1, DRT=   89, UNIT=    0, TYPE=    0, SUBTYPE=    9
 SECTOR % 0
   0: SYSTEM DISC ....                100: ................
  10: 3000MH7925U0....                110: ................
  20: ................                120: ................
  30: ................                130: ................
  40: ................                140: ................
  50: ................                150: ................
  60: ................                160: ................
  70: ................                170: ................
```

```
ENTER ADDRESS: 0,1;0
SECTOR % 0
  0: 051531 051524 042515 020104 044523 041440 000011 001007
 10: 031460 030060 046510 033471 031065 052460 000000 000000
 20: 000000 000000 000000 000000 000000 000000 000000 000000
 30: 000000 000000 000000 000000 000000 000000 000000 000000
 40: 000000 000000 000000 000000 000000 000000 000000 000000
 50: 000000 000000 000000 000000 000000 000000 000000 000000
 60: 000000 000000 000000 000000 000000 000000 000000 000000
 70: 000000 000000 000000 000000 000000 000000 000000 000000
100: 000000 000000 000000 000000 000000 000000 000000 000000
110: 000000 000000 000000 000000 000000 000000 000000 000000
120: 000000 000000 000000 000000 000000 000000 000000 000000
130: 000000 000000 000000 000000 000000 000000 000000 000000
140: 000000 000000 000000 000000 000000 000000 000000 000000
150: 000000 000000 000000 000000 000000 000000 000000 000000
160: 000000 000000 000000 000000 000000 000000 000000 000000
170: 000000 000000 000000 000000 000000 000000 000000 000000

ENTER ADDRESS: 28,1;0
SECTOR % 34
  0: 000056 000136 000026 000020 043600 173010 030263 026674
 10: 026674 000011 001007 001171 000000 000136 000000 032600
 20: 000003 000020 000000 013716 013560 024000 000000 000005
 30: 000264 000020 000000 002004 000000 000131 000000 033040
 40: 000000 033120 000000 033320 000400 033530 000400 033540
 50: 000000 013723 000000 013726 000003 000003 000220 036617
 60: 000000 032262 000200 042300 000000 032251 002000 040300
 70: 000000 032252 000200 042500 000000 032276 000400 042700
100: 000000 032277 000007 036610 000000 071750 000060 037037
110: 000000 071675 000170 037117 000000 071701 000055 037307
120: 000000 071713 000106 037554 000000 071741 000170 037364
130: 000000 071727 030263 043600 000000 071770 003544 174200
140: 000000 000004 014100 160100 000000 013731 025744 132100
150: 000000 014012 004120 125700 000000 014142 005060 120600
160: 000000 014163 003734 114600 000000 014210 006264 106300
170: 000000 014230 003544 102500 000000 014262 003024 077400

ENTER ADDRESS: %71741,1;0
SECTOR % 71741
  0: 002016 001010 000004 001007 000000 000000 000000 000000
 10: 000000 000000 000000 000000 000000 000000 046510 033471
 20: 031065 052460 000000 000000 000000 000000 000000 032600
 30: 000000 024000 000410 000000 000000 000000 000000 000000
 40: 000000 000000 000000 000000 000000 000000 000000 000000
 50: 000000 000000 000000 000000 000000 000000 000000 000000
 60: 000000 000000 000000 000000 000000 000000 000000 000000
 70: 000000 000000 000000 000000 000000 000000 000000 000000
100: 000000 000000 000000 000000 000000 000000 000000 000000
110: 000000 000000 000000 000000 000000 000000 000000 000000
120: 000000 000000 000000 000000 000000 000000 000000 000000
130: 000000 000000 000000 000000 000000 000000 000000 000000
140: 000000 000000 000000 000000 000000 000000 000000 000000
150: 000000 000000 000000 000000 000000 000000 000000 000000
160: 000000 000000 000000 000000 000000 000000 000000 000000
170: 000000 000000 000000 000000 000000 000000 000000 000000
```

```
 ENTER ADDRESS:
ENTER FUNCTION: EDIT
 >MODIFY 0,7,1
  SECTOR % 0
    7: 001007:=000000
  WRITTEN
 >MODIFY 28,%12,1
  SECTOR % 34
   12: 001007:=000000
  WRITTEN
 >MODIFY 28,%32,1
  SECTOR % 34
   32: 000000:=000000
  WRITTEN
 >MODIFY %71741,1,1
  SECTOR % 71741
    1: 001010:=000000
  WRITTEN
 >
ENTER FUNCTION: STOP
END OF PROGRAM.
Enter Your Program Name (type HELP for program information)
:
->START
IS IT OK TO ABORT SYSTEM (Y OR N)?Y
)
HP32033C.F0.D3
WHICH OPTION (WARMSTART/COOLSTART)? COO
ANY CHANGES?
**WARNING** DEFAULT VIRTUAL MEMORY SIZES BEING USED

DATE (M/D/Y)?1/3/82
TIME (H:M)?16:36
SUN, JAN  3, 1982,  4:36 PM? (Y/N)Y
LOG FILE NUMBER 634 ON
*WELCOME*
:HELLO OPERATOR.SYS;HIPRI
16:36/12/SP#6/SPOOLED OUT
16:36/#S1/13/LOGON FOR: OPERATOR.SYS,OPERATOR ON LDEV #20
HP3000 / MPE IV C.F0.D3.  SUN, JAN  3, 1982,  4:36 PM
:
```

APPENDIX C

The following is a sample dialog using SLEUTHSM to zero out the volume label on a disc. The dialog for SLEUTH, used on Series II/III machines, is similiar. The method of loading the diagnostic is different, since SLEUTHSM is run as a program under AID, while SLEUTH is a standalone diagnostic.

```
->LOAD
%p%

Diagnostic/Utility System Revision 01.01
Enter Your Program Name (type HELP for program information)
:AID

AID 01.01

>  10 LOAD SLEUTHSM
Program Loaded!!
The Next Available Statement Number is
>5000 DEV 0,11,1,100,0
>5010 DB AA,128,0
>5020 WD 0,AA(0),3,0,0,0
>5030 END
>5040 RUN

End of AID user program
>5040 EXIT
Confirm you want to ERASE the current program(Y or N)?Y


Diagnostic/Utility System Revision 01.01
Enter Your Program Name (type HELP for program information)
:SADUTIL
Disc Utility A01.03 (C) Hewlett-Packard Co., 1976
LIST LOGICAL DEVICES?
DISC CONFIGURATION CHANGES? Y
LOGICAL DEVICE? 1
DRT? 89
UNIT? 0
TYPE? 0
SUBTYPE? 9
LOGICAL DEVICE?
LIST LOGICAL DEVICES? N
SERIAL DEVICE CHANGES? Y
DRT? 73
UNIT? 0
TYPE? 24
SUBTYPE? 0
SYSTEM DISC UNINITIALIZED
```

```
ENTER FUNCTION: PDSK 1
 ENTER ADDRESS: 0,1;A
 LDEV=   1, DRT=  89, UNIT=    0, TYPE=    0, SUBTYPE=    9
 SECTOR % 0
   0: 000000 000000 000000 000000 000000 000000 000000 000000
  10: 000000 000000 000000 000000 000000 000000 000000 000000
  20: 000000 000000 000000 000000 000000 000000 000000 000000
  30: 000000 000000 000000 000000 000000 000000 000000 000000
  40: 000000 000000 000000 000000 000000 000000 000000 000000
  50: 000000 000000 000000 000000 000000 000000 000000 000000
  60: 000000 000000 000000 000000 000000 000000 000000 000000
  70: 000000 000000 000000 000000 000000 000000 000000 000000
 100: 000000 000000 000000 000000 000000 000000 000000 000000
 110: 000000 000000 000000 000000 000000 000000 000000 000000
 120: 000000 000000 000000 000000 000000 000000 000000 000000
 130: 000000 000000 000000 000000 000000 000000 000000 000000
 140: 000000 000000 000000 000000 000000 000000 000000 000000
 150: 000000 000000 000000 000000 000000 000000 000000 000000
 160: 000000 000000 000000 000000 000000 000000 000000 000000
 170: 000000 000000 000000 000000 000000 000000 000000 000000

 ENTER ADDRESS:
ENTER FUNCTION: STOP
END OF PROGRAM.
Enter Your Program Name (type HELP for program information)
:
->LOAD
IS IT OK TO ABORT SYSTEM (Y OR N)?Y
)
HP32033C.F0.D3
WHICH OPTION (COLDSTART/RELOAD/UPDATE)? REL
WHICH OPTION (SPREAD/COMPACT/RESTORE/ACCOUNTS/NULL)? ACC
ANY CHANGES?
NON-SYSTEM VOLUME ON LDEV 1
ADD TO SYSTEM VOLUME SET? Y
ENTER VOLUME NAME? MH7925U0
LOGICAL PACK SIZE IN CYLINDERS = 815.?
SUSPECT TRK  LDEV #1 CYL=23 HEAD=6 (SECTORS %32500-%32577)
DELETE OR REASSIGN? DEL
BANK 0 DEPENDENT MEMORY USED - 26884

DATE (M/D/Y)?1/3/82
TIME (H:M)?16:%57
SUN, JAN  3, 1982,  4:57 PM? (Y/N)Y
LOG FILE NUMBER 633 ON
*WELCOME*
:HELLO OPERATOR.SYS;HIPRI
16:57/12/SP#6/SPOOLED OUT
16:57/#S1/13/LOGON FOR: OPERATOR.SYS,OPERATOR ON LDEV #20
HP3000 / MPE IV C.F0.D3.  SUN, JAN  3, 1982,  4:57 PM
:FILE T;DE=VTAPE!!!
FILE T;DEV=TAPE
:RESTORE *T;@,@.@;OLDDATE
?16:57/#S1/13/I.DEV# FOR "T" ON TAPE (NUM)?
=REPLY 13,7
16:57/9/VOL UNLABELLED   MOUNTED ON LDEV# 7
```

```
FILES RESTORED = 186

FILES NOT RESTORED = 25

   FILE      .GROUP    .ACCOUNT FILESET     REASON

   CATALOG   .PUB      .SYS          1      BUSY
   CONFDATA  .PUB      .SYS          1      BUSY
   DEVREC    .PUB      .SYS          1      BUSY
   HIOLPRT0  .PUB      .SYS          1      BUSY
   HIOMDSC1  .PUB      .SYS          1      BUSY
   HIOTAPE0  .PUB      .SYS          1      BUSY
   HIOTERM0  .PUB      .SYS          1      BUSY
   ININ      .PUB      .SYS          1      BUSY
   INITIAL   .PUB      .SYS          1      BUSY
   LOAD      .PUB      .SYS          1      BUSY
   LOADMAP   .PUB      .SYS          1      BUSY
   LOG       .PUB      .SYS          1      BUSY
   MAKECAT   .PUB      .SYS          1      BUSY
   MEMLOGP   .PUB      .SYS          1      BUSY
   PFAIL     .PUB      .SYS          1      BUSY
   PROGEN    .PUB      .SYS          1      BUSY
   PVPROC    .PUB      .SYS          1      BUSY
   SDFCHECK  .PUB      .SYS          1      BUSY
   SDFCOM    .PUB      .SYS          1      BUSY
   SDFGEN    .PUB      .SYS          1      BUSY
   SDFLOAD   .PUB      .SYS          1      BUSY
   SEGDVR    .PUB      .SYS          1      BUSY
   SEGPROC   .PUB      .SYS          1      BUSY
   SYSDUMP   .PUB      .SYS          1      BUSY
   UCOP      .PUB      .SYS          1      BUSY

:BYE

CPU=45. CONNECT=9. SUN, JAN  3, 1982,  5:06 PM
17:06/#S1/13/LOGOFF

:
=SHUTDOWN


HALT  0
->
```

APPENDIX D

The following is a sample dialog showing how SADUTIL can be used to list and recover disc files. **These are then restored using RECOVER2.**

```
=SHUTDOWN

SESSION ABORTED BY SYSTEM MANAGEMENT
CPU=1. CONNECT=1. SUN, JAN  3, 1982,  4:36 PM
16:36/#S1/13/LOGOFF
16:36/1/ALL JOBS LOGGED OFF
SHUT

HALT 15
->LOAD
*^*

Diagnostic/Utility System Revision 01.01
Enter Your Program Name (type HELP for program information)
:SADUTIL
Disc Utility A01.03 (C) Hewlett-Packard Co., 1976
LIST LOGICAL DEVICES? N
DISC CONFIGURATION CHANGES? Y
LOGICAL DEVICE? 1
DRT? 89
UNIT? 0
TYPE? 0
SUBTYPE? 9
LOGICAL DEVICE?
LIST LOGICAL DEVICES? N
SERIAL DEVICE CHANGES? Y
DRT? 73
UNIT? 0
TYPE? 24
SUBTYPE? 0
DRT  #89, UNIT #0 NOT READY
DRT  #89, UNIT #0 NOT READY
ENTER FUNCTION: PFIL
 ENTER NAME: @.PUB.GOERTZ

 ACCOUNT = GOERTZ    GROUP = PUB

 BADLABEL  BANNER    COPYLIB   COPYLIBK  CRASH     CRASH2
 CRASH2P   CRASHP    CRASHU    DEBUFFER  DRWIZARD  DECOM3
 DECOMP6   DIRMATCH  DISCADDR  DSCAN     DSCANTST  DUANE
 DWNTST    DWNTSTP   ENTRYPNT  EVERGRN   EXAMPLE   EXAMPLEP
 FLABEL    FLIMIT    FLUTIL3   FORTRAN   FTNLIST   FTNNEW
 FTNUSL    GETSTRNG  GRDSCHMA  GRDTEST   GRDTSTP   ID
 IDGGEN    KSAMRBLD  LABJOB    LIBREST   LIMCHNG   LSTALLOC
 PICS      PICS2     PICTEST   PICTEST1  PRINTER   QDISPLAY
 RTMX      SEGPROG   SEP329    SETCOBOL  SETTDPC   SL
 SL2       SLPMAP    SLPMAPP   SLPMAPQ   SM        SOO
 SPD4800   SPL       SPL2      SPLLAB    SPLLAB2   SPLLAB3
 SPLSTD    SPLXREF   SUSTRACK  SWITCH    TAPELAB   TERMID
 TEST      TESTER    TESTFILE  TESTVM    UDC       UDCCPL
 UDCUTIL   ULDSET    USERINIT  UT817     XXX       XXXP
 XYZP
```

```
ENTER NAME: BANNER.PUB.GOERTZ,1

ACCOUNT = GOERTZ    GROUP = PUB

BANNER    1    %73143

ENTER NAME: BANNER.PUB.GOERTZ,2

ACCOUNT = GOERTZ    GROUP = PUB

BANNER      12/ 1/81   12/ 1/81   12/ 3/81
```

```
ENTER NAME:
ENTER FUNCTION: SAVE
READY SERIAL DEVICE FOR WRITE
FILE NAME (OR LDEV#,%SECTOR ADDRESS)? 1,%73067
BADLABEL.PUB      .GOERTZ    - CONTENTS OF LABEL
RETRIEVE THIS FILE (Y/N)? Y
BADLABEL.PUB      .GOERTZ    1    73067
FILE NAME (OR LDEV#,%SECTOR ADDRESS)? BANNER.PUB.GOERTZ
DATE?
BANNER   .PUB     .GOERTZ    1  %73143
FILE NAME (OR LDEV#,%SECTOR ADDRESS)? COPYLIB.PUB.GOERTZ
DATE?
COPYLIB .PUB      .GOERTZ    1  %73400
FILE NAME (OR LDEV#,%SECTOR ADDRESS)? 1,%75754
COPYLIBK.PUB      .GOERTZ    - CONTENTS OF LABEL
RETRIEVE THIS FILE (Y/N)? Y
COPYLIBK.PUB      .GOERTZ    1    75754
FILE NAME (OR LDEV#,%SECTOR ADDRESS)?
ENTER FUNCTION: STOP
END OF PROGRAM.
Enter Your Program Name (type HELP for program information)
:
->SSTART
INVALID - USE HELP

Invalid Command or Input
Enter Your Program Name (type HELP for program information)
:
->START
IS IT OK TO ABORT SYSTEM (Y OR N)?Y
)
HP32033C.F0.D3
WHICH OPTION (WARMSTART/COOLSTART)? COO
ANY CHANGES?

DATE (M/D/Y)?1/3/82
TIME (H:M)?16:43
SUN, JAN  3, 1982,  4:43 PM? (Y/N)Y
LOG FILE NUMBER 635 ON
*WELCOME*
```

```
:HELLO OPERATOR.SYS;HIPRI
16:43/12/SP#6/SPOOLED OUT
16:43/#S1/13/LOGON FOR: OPERATOR.SYS,OPERATOR ON LDEV #20
HP3000 / MPE IV C.F0.D3.  SUN, JAN  3, 1982,  4:43 PM
:HELLO MANAGER.SYS

CPU=1. CONNECT=1. SUN, JAN  3, 1982,  4:43 PM
16:43/#S1/13/LOGOFF

16:43/#S2/14/LOGON FOR: MANAGER.SYS,PUB ON LDEV #20
HP3000 / MPE IV C.F0.D3.  SUN, JAN  3, 1982,  4:43 PM
:RUN RECOVER2.PUB.SYS

RECOVER2 C00.00 (C) HEWLETT-PACKARD CO., 1976
WISH TO KEEP EXISTING COPIES OF FILES (Y/N)?N
?16:43/#S2/15/LDEV# FOR "RECOVTP" ON TAPE (NUM)?
=REPLY 15,7
16:43/9/VOL UNLABELLED   MOUNTED ON LDEV# 7
BADLABEL.PUB      .GOERTZ    1   00100072733
BANNER   .PUB     .GOERTZ    1   00100073010
COPYLIB  .PUB     .GOERTZ    1   00100276217
COPYLIBK.PUB      .GOERTZ    1   00100073071
IS THERE ANOTHER RECOVERY TAPE (Y/N)? N

END OF PROGRAM

:BYE

CPU=9. CONNECT=2. SUN, JAN  3, 1982,  4:44 PM
16:44/#S2/14/LOGOFF
```