Archiving Data from Gulf Coast Regional Aquifer System Analysis Study

By Kimberly A. Kirkpatrick

U.S. GEOLOGICAL SURVEY
Open-File Report 92-661

Portland, Oregon

1993

U. S. DEPARTMENT OF THE INTERIOR
MANUEL LUJAN, JR., Secretary
U.S. GEOLOGICAL SURVEY
DALLAS L. PECK, Director

For additional information
write to:
District Chief
U.S. Geological Survey, WRD
10615 S.E. Cherry Blossom Drive
Portland, Oregon 97216

Copies of this report can
be purchased from:

U.S. Geological Survey
Books and Open-File Reports Section
Federal Center, Box 25425
Denver, Colorado 80225

CONTENTS

ARCHIVING DATA FROM GULF COAST REGIONAL AQUIFER SYSTEM ANALYSIS STUDY

By Kimberly A. Kirkpatrick

Abstract

A pilot project was conducted by the U.S. Geological Survey to develop a methodology to store Regional Aquifer System Analyses study data in an archival format for the purpose of permanent storage and simple retrieval.

A 900-megabyte, write-once, read-many drive was purchased and connected to a Data General AViiON 300 workstation[1]. Twelve write-once, read-many disks also were purchased to allow for experimentation and for producing ten copies of all pertinent Gulf Coast Regional Aquifer System Analyses study data.

Write-once, read-many, optical-disk technology offer improvements over printed paper or magnetic disks and tapes for archival storage and retrieval, because the disks require minimal storage space, do not require temperature- or humidity-controlled environments, do not require a manual process for retrieval, and can remain unaltered for long time periods.

INTRODUCTION

Background

The U.S. Geological Survey (USGS) has conducted 25 studies in the Regional Aquifer System Analysis Program (RASA) since its inception in 1978. Each study collected and stored data such as water levels, hydraulic conductivity, water use (pumpage), aquifer geometry, aquifer properties, and other geologic and hydrologic information. In addition, the data were organized into input sets for simulation modeling, and the results of these simulations were stored as well. Seventeen of the studies have been completed, with the remainder to be completed by 1996.The large amounts of RASA data require a means of permanent storage and simple retrieval by hydrologists inside and outside of the USGS. Much of the RASA data has become infrequently used upon completion of the studies and is stored in varying forms in many different USGS offices. The media used for storage of data from completed projects differs by project and by type of data. Much of the data remains on magnetic tape, magnetic disks on the USGS's Amdahl computer in Reston, Virginia, magnetic disks on Prime minicomputers in field offices, and on printed paper. Retrieval of inactive, archived information stored in any of these forms is a cumbersome and costly manual process. The difficulty of data retrieval is one of the main reasons RASA data are inactive.

---

[1] Use of brand names in this report is for identification purposes only and does not constitute endorsement by the U.S. Geological Survey.

Use of write-once, read-many (WORM) optical-disk technology will greatly increase data usability and decrease data storage and labor costs as compared with conventional, manual storage and retrieval methods, because the disks require minimal storage space and do not require temperature- and humidity-controlled environments. Contents of paper and magnetic media can be stored on the WORM disks as digital text data.

Purpose and Scope

This report describes the methodology used to store data in an archival format for the Gulf Coast RASA study, 1983-1990. Specifically, it describes methods used to: (1) transfer data and accumulate data at a central site in digital form; (2) internally document each file in a standard form; (3) store an index to the data internally on the WORM media; (4) store the RASA study data on a WORM disk and verify that data; and (5) replicate copies of the pertinent data.

The scope of the project did not allow mass distribution of data; ten copies of all pertinent data from the Gulf Coast RASA study were produced.

ALTERNATIVES FOR DATA ARCHIVING

Paper

Printed paper can remain unaltered for time periods longer than ten years; however, it does require a humidity- and temperature-controlled environment. Paper requires a large amount of storage space when compared with WORM disks, making it an inefficient archival medium. The capacity of one side of a 900-megabyte WORM disk is approximately equal to 8,000 pieces of paper. In addition, paper does require a manual retrieval process.

Magnetic Tapes/Disks

Magnetic disks have failure ratings from 1 to 3 years, and magnetic tape can deteriorate in as little as 3 years. Magnetic media also require humidity- and temperature-controlled environments similar to paper. In comparison to WORM disks, it would take eight, 2,400-foot reels of magnetic tape to hold as much data as a 900 megabyte WORM disk.

Write-Once-Read-Many (WORM) Disks

WORM disks do not require humidity- or temperature-controlled environments, and the storage space required for WORM disks is minimal. A single, 5.25-inch disk can contain 900 megabytes of information, has an average seek time of 90 milliseconds, has a sustained data transfer rate of 2.6 - 5.2 mega-bytes per second, has an error rate less than 10^{-12}, and is expected to have an archival life of nearly 40 years (Toshiba America Information Systems, Inc., 1989).

A Data General Optical Drive (Toshiba Model #WM-D070-AC) was used for this study. At the present time, no federal or international standards governing the format of 5.25-inch WORM disks exist. In general, a WORM disk may be written to and read from on WORM drives from only one vendor. However, European Computer Manufacturer's Association (ECMA) has developed ECMA-167, a standard governing WORM File Systems, and is presently working on an International Standards Organization (ISO) standard (Hume, September/ October 1992).

Use of WORM technology will greatly increase data usability and decrease data storage and labor costs as compared to conventional, manual storage and retrieval methods. This makes it the best alternative for an archival medium.

APPROACH

Data Transfer

Much of the data from the Gulf Coast RASA study remains on magnetic disks on a Prime minicomputer and on magnetic tape at the USGS office in Austin, Texas. One goal of this study was to transfer and accumulate the data files at a central site in digital form. The destination site for these data was a Data General workstation in Portland, Oregon. To accomplish this, the Prime utility *ftr* (File Transfer Request) was used to transfer the data files from the Prime minicomputer in Austin, Texas, to the Prime minicomputer in Portland, Oregon. The syntax for the *ftr* command is:

ftr source-pathname destination-pathname *-dstn_site* sitename

An example of a procedure using *ftr* from a Prime minicomputer follows:

ftr l.saltdomes fts_depot>== *-dstn_site* dorprn

As a result, the file, *l.saltdomes*, would be transferred to the *fts_depot* directory on *dorprn* (the Oregon District's Prime minicomputer). When transferring a large number of files, it is possible to use the wildcard, @@, in place of the source-pathname with this command, rather than transfer the files one at a time. Because of the location of the data files, it was necessary to accomplish the data transfer in the above manner. Depending on the location of the data files, there may exist different approaches to get the data to a Data General workstation.

All of the data files from the Gulf Coast RASA study originally were stored in the form of P-STAT (a statistical analysis software package) libraries. One requirement for the archive was to store the data files in such a manner that they would be independent of proprietary software. To make "generic" files for use with existing and future computer equipment, a program (Appendix A) was written to transform P-STAT libraries to ASCII (American Standard Code for Information Interchange) data files. The program inserts 100 blank lines into the top of each file to allow later manual entry of the file description. An example of how to run the program on a Prime minicomputer follows:

r archive l.saltdomes

The name of the program is *archive*, the name of the input file is *l.saltdomes*, and the name of the resulting output file is *saltdomes.pqz*.

Once the file conversion was completed, the Unix utility *ftp* (File Transfer Program) was used to transfer the data files from the Prime minicomputer to a Data General workstation. An example of a procedure using *ftp* to transfer all files ending with *.pqz* from a directory on a Prime minicomputer and the directory */archive* on a Data General workstation follows:

ftp dorprx

cd /archive

mput @@.pqz

quit

The first command opens a connection with a remote system, and prompts the user for a username and password. The name of the workstation where the project disk partition resides is *dorprx*. The second command changes the working directory to */archive* (the project disk partition). The third command allows the user to send a number of files to the current remote directory. By default, *mput* prompts the user for verification before sending a file. In this case, the wild-card symbol, @@, was used to allow the user to transfer all files with a *.pqz* extension. The fourth command allows the user to terminate the *ftp* session and return to the Primos command line.

File Documentation

Owing to the large amounts of RASA data associated with this archive, it was desirable to maintain the files in a uniform manner. For this study, each file was internally documented in a standard form. The file descriptions for this project were written by the Gulf Coast RASA Project Chief, Alex K. Williamson. Each description briefly summarized the contents of the data file and was inserted into the first 100 lines of the data file using *vi*, a Unix screen editor.

Indexing

To help locate specific data files, two index files describing the overall structure of the archived RASA data were constructed. One index file was in the form of a Unix directory listing. It lists the files exactly as they would appear if the contents of the working directory were displayed using the Unix command *ls -laR*. The second index file contains the same information organized in a different manner. There are seven headings (defined by the Gulf Coast RASA Project Chief) under which the information was divided. For example, one heading is titled *Model Output Directories*. All data files associated with model output were located under this heading. Both index files have been stored on the WORM disk on the first logical tape with the data files to assist retrieval and determining the organization of the Gulf Coast RASA study data.

Storage and Replication of Data

After completing the file documentation and listings, the Bourne shell script, make_tape (Appendix B), was used to archive the RASA study data to the WORM disk. An example of this process follows:

```
dorprx% make_tape

This program allows a tar tape to be created containing selected directories.

Mount the tape and type in the name of the tape device.
For example: /dev/pmt/0  (Default: /dev/rmt/1)
/dev/wmt/1

Is the tape mounted and ready [y/n]?
y

Which logical tape is this? You must start your numbering with '0'.
0

Which directory would you like to tar? You must specify the full
pathname. (Example: /rasadata/archive )
/archive/directory
a archive.cpl 5 tape blocks
```

a load_tape 10 tape blocks

a make_tape 8 tape blocks
a readme.descriptions 119 tape blocks
a readme.structure 104 tape blocks

Do you want to tar another directory? [y/n]
y

Which logical tape is this? You must start your numbering with '0'.
1

Which directory would you like to tar? You must specify the full
pathname. (Example: /rasadata/archive)
/archive/cores
a CoreDesc 7 tape blocks
a pi4all.pqz 59392 tape blocks

Do you want to tar another directory? [y/n]
n
dorprx%

This script prompts for the name of the tape device, the logical tape (the data is stored on the WORM disk as logical tapes), and the directory to be archived. This process continues until the script is terminated. This script can be used with any magnetic tape devices (8mm tapes, 9-track tapes, etc.) or logical tape devices, such as a WORM disk drive, as long as the name of the tape device is known. Even though a WORM disk drive was used for this project, Data General's software treats the WORM drive as if it were a tape drive. When this script refers to tapes, it is referring to the WORM disk. When using WORM disks, this script will return a "tar write error" if the same logical tape is input twice. A "tar write error" does not occur on other types of tapes, such as 8mm tapes, that have the capability of being written to more than once. In this case, the script will allow any data previously saved on that logical tape to be overwritten. For this project, it took approximately three hours to archive all of the pertinent Gulf Coast RASA study data, nearly 370 megabytes, from magnetic disk to a WORM disk platter.

The Bourne shell script, load_tape (Appendix C), was used to restore the RASA study data from the WORM disk. An example of this procedure follows:

mcmcdorprn% load_tape

Gulf Coast RASA Data

This program will allow the restoration of Gulf Coast RASA Study data
from the write-once, read-many disk.

Mount the tape and type in the name of the tape device.
For example: /dev/pmt/0 (Default: /dev/rmt/1)
/dev/wmt/0

Is the tape mounted and ready [y/n]?
y

Gulf Coast RASA Data

Logical Tape	Name of Directory	Size
1	directory	273 blocks
2	cores	59432 blocks
3	kuiper	11051 blocks
4	logs	7022 blocks
5	misc	105592 blocks
6	pisub5	266 blocks
7	qw	182145 blocks
8	rasadata	175291 blocks
9	watlevels	108626 blocks
10	wgcrasa	44565 blocks
11	williamson	212 blocks

Which logical tape would you like to restore? (1-11)
11

Which directory would you like to restore to? You must specify the full pathname. (Example: /rasadata/archive)
/tmp/temp
tar: blocksize = 32
x WillDesc, 5962 bytes, 12 tape blocks
x pstat/pop/wgcpop30.pqz, 22711 bytes, 45 tape blocks
x pstat/pop/wgcpopyr.pqz, 61503 bytes, 121 tape blocks
x subsid/oilsub.pqz, 6752 bytes, 14 tape blocks

Do you want to restore another logical tape [y/n]?
y

Gulf Coast RASA Data

Logical Tape	Name of Directory	Size
1	directory	273 blocks
2	cores	59432 blocks
3	kuiper	11051 blocks
4	logs	7022 blocks
5	misc	105592 blocks
6	pisub5	266 blocks
7	qw	182145 blocks
8	rasadata	175291 blocks
9	watlevels	10626 blocks
10	wgcrasa	44565 blocks
11	williamson	212 blocks

```
Which logical tape would you like to restore? (1-11)
6

Which directory would you like to restore to? You must specify the
full pathname. (Example: /rasadata/archive )
/tmp/temp
tar: blocksize = 32
x Pisub5Desc, 2153 bytes, 5 tape blocks
x gcprhd.agg.pqz, 126856 bytes, 248 tape blocks

Do you want to restore another logical tape [y/n]?
n
mcmcdorprn%
```

This script prompts for the name of the tape device, the logical tape to be restored, and the destination directory that will receive the retrieval. This process continues until the script is terminated. If this script is used for other projects similar to this one, the table of contents entries in this script will need to be replaced with the table of contents from the other project.

Ten copies of all pertinent Gulf Coast RASA study data were produced utilizing the above Bourne shell scripts.

SUMMARY

The large amounts of data generated by RASA studies require a uniform means of permanent storage and simple retrieval by hydrologists inside and outside of the USGS. Use of WORM technology for this project greatly increased data usability and decreased data storage and labor costs when compared to the other conventional methods, such as printed paper or magnetic disks and tapes.

The difficulty of data retrieval was one of the main reasons that the Gulf Coast RASA data were inactive. Through the file documentation, indexing, and retrieval process, many of the problems have been alleviated. A menu-driven script assists in the retrieval of data and consulting the file listings and internal documentation within each file will help to determine the organization of the Gulf Coast RASA study data.

SELECTED REFERENCES

Hume, Andrew, September/October 1992, Report on ANSI X3B11.1: WORM File Systems: Login, v. 17, no. 10, p. 39-40.

McLeod, Jonah, 1988, Optical storage may fulfill its promise - after years of trying: Electronics, v. 61, no. 10, p. 75.

Rosch, Winn L., 1987, Backup choices from tapes to disks to WORMs: PC Magazine, v. 6, no. 12, p. 102-108.

_____ 1987, WORMs for mass storage: PC Magazine, v. 6, no. 12, p. 135-148.

Stephens, Mark, 1987, WORM disc drive features 800 megabyte storage capacity: Infoworld, v. 9, no. 126, p. 19.

Toshiba America Information Systems, Inc., Disk Products Division, 1989, WM-070 Series Specifications + Characteristics.

APPENDIXES

APPENDIX A

Program to convert PSTAT libraries to ASCII data files
Written by Alex K. Williamson

```
/ Script Name: Convert from PSTAT to ASCII files
/ Author: Alex K. Williamson
/ Purpose: To allow a user to convert from PSTAT libraries to ASCII files
/ Modified: By Kimberly Kirkpatrick, 10/14/92

/*************************************************************************
/* This part sets up the temporary variables, destination files, checks to see
/* if the source files exist, and inserts a header line.
/*************************************************************************

&ARGS LIBNAME:TREE = L.REG1ALL ; SUBDIR:= Z
 &IF [NULL %LIBNAME%] &THEN &STOP  &MESSAGE NO LIBNAME
 &IF ^ [EXISTS %LIBNAME% -FILE] &THEN &STOP  &MESSAGE NO FILE NAMED %LIBNAME%
&SEVERITY &ERROR &ROUTINE IF_ERR
&ON ERROR &ROUTINE IF_ERR
&S DATIME  := [SUBSTR [DATE -FTAG] 3 9 ]
&S DSNAME := [AFTER %LIBNAME% L. ]
&S OUTNAM1  := [QUOTE T$1.%DSNAME% ]
&S OUTNAM2  := [QUOTE T$2.%DSNAME% ]
&S FINLNAME  := PI>ARCHIVE>%SUBDIR%>%DSNAME%.PQZ
DELETE (%OUTNAM1% %OUTNAM2%) -FORCE -NQ -NVFY
DELETE %FINLNAME% -FORCE -NQ -NVFY
TYPE ********** PSTAT LIBRARY DUMP TO ASCII OF [PATHNAME %LIBNAME%] **********


/*************************************************************************
/* This part runs the program "pstat.run", and inserts the "outnam1" file,
/* which contains file data.
/*************************************************************************

&DATA R  PSTAT>PSTAT.RUN
 BATCH $
  PRIMOS [QUOTE DATE          ], NO.JOB, PR %OUTNAM1% $
  PRIMOS [QUOTE LD %LIBNAME% -DET ], NO.JOB, PR %OUTNAM1% $
ATTACH %DSNAME%, LIB %LIBNAME%$
MODIFY %DSNAME%, DES %DSNAME%.DES$
L %DSNAME%.DES , PR %OUTNAM1% ,
            GAP 1, SKIP 5,TITLES $
LIST  %DSNAME%, DATA.ONLY, MARGIN 0,
          PR %OUTNAM2% ,
            GAP 1, NO DASHES   $
ERASE.EVERYTHING$  /* TO MAKE COMPATIBLE WITH OLD COMMAND FILES
 END $
&END
CLOSE L.@@
&DATA ED %OUTNAM1%
```

APPENDIX A--Continued

Program to convert PSTAT libraries to ASCII data files

```
/*************************************************************************
/* This part inserts the following text lines, 100 ":/*" lines to allow
/* room for a file description, and two text lines denoting the "end of
/* notes" and the beginning of the file data.
/*************************************************************************

I ********** PSTAT LIBRARY DUMP TO ASCII OF [PATHNAME %LIBNAME%] **********
BOTTOM
I
FIRST # IN TABLE ABOVE = COLUMN # OF VARIABLE AS LISTED IN ASCII FILE BELOW
M1,M2,M3, REPRESENT MISSING DATA OF 3 DIFFERENT TYPES OR SOURCES
BELOW ARE LISTED SOME MANUALLY ENTERED NOTES, PADDED WITH BLANK LINES
  SO THAT THE DATA BEGINS ON LINE NUMBER 101
:/*
:/*  (**Repeat this line 98 times for a total of 100 ":/*" lines.**)

PO99
D99
I ************************** END   OF NOTES ************************
I THE ASCII DATA FILE FOLLOWS THIS LINE:  **************************


/*************************************************************************
/* This part inserts the data from "outnam2", deletes the two files "outnam1"
/* and "outnam2", and terminates.
/*************************************************************************

LOAD  %OUTNAM2%
FILE  %FINLNAME%
&END
DELETE (%OUTNAM1% %OUTNAM2%) -FORCE -NQ -NVFY
&RETURN
&ROUTINE IF_ERR
&RETURN
```

APPENDIX B

Program to make a tar tape of the Gulf Coast RASA Study Data
Written by Kimberly A. Kirkpatrick

```
# ! /bin/sh
# Script Name: Make Archive Tape
# Author: Kimberly Kirkpatrick
# Purpose: To allow a user to make a tar tape for the RASA Study
# Date: September 29, 1992


#*************************************************************************
#* This procedure assigns /dev/rmt/1 as the default tape drive. It prompts
#* the user to enter another name if the default is not suitable. It then
#* prompts to see if the tape is mounted and ready before continuing.
#*************************************************************************
tape_device()
{
name="/dev/rmt/1"
echo " "
echo " "
echo "This program allows a tar tape to be created containing selected directories."

while :
do
  echo " "
  echo "Mount the tape and type in the name of the tape device."
  echo "For example:  /dev/pmt/0  (Default: /dev/rmt/1)"
  read namet

  if [ "$namet" != "" ]
  then
    name="$namet"
  fi

  if test -c "$name"
  then
    break
  else
    echo " "
    echo "$name does not exist.  Please check, and try again."
  fi
done

while :
do
  echo " "
  echo "Is the tape mounted and ready [y/n]?"
  read answer
```

Program to make a tar tape of the Gulf Coast RASA Study Data

```
  if [ "${answer}" = "y" ] || [ "${answer}" = "yes" ]
  then
    mt -f ${name}n rew
    break
  else
        echo "Please mount the tape."
        echo ""
  fi
done
}

#*****************************************************************************
#* This procedure rewinds the tape.
#*****************************************************************************
rewind_tape()
{
mt -f ${name} rew
}

#*****************************************************************************
#* This procedure prompts the user to enter the full pathname of the
#* directory to be archived.  A check is done to make sure the specified
#* pathname is not a file, that it exists, and that the user has read
#* privileges for the directory.  If so, the directory contents are copied
#* to tape.
#*****************************************************************************
directory()
{
while :
do
  echo " "
  echo "Which directory would you like to tar?  You must specify the full"
  echo "pathname.  (Example:  /rasadata/archive )"
  read dir

  if test -f "$dir"
  then
    echo " "
    echo "$dir is a file, not a directory"
  elif test -r "$dir"
  then
    cd ${dir}
    tar cvf ${name}n *
    break
  else
    echo " "
    echo "You do not have the necessary privileges for the $dir directory,"
    echo "or it does not exist.  Please check, and try again."
```

APPENDIX B--Continued

Program to make a tar tape of the Gulf Coast RASA Study Data

```
  fi
done
}

#*****************************************************************************
#* This procedure prompts the user to enter a logical tape number and file
#* steps forward the appropriate number.
#*****************************************************************************
logical_tape()
{
echo " "
echo "Which logical tape is this?  You must start your numbering with '0'."
read tape
mt -f ${name}n fsf ${tape}
}

#*****************************************************************************
#* This is the main body of the program.  The user is prompted for the tape
#* device, the logical tape number, the full pathname of the directory, and
#* if they would like to tar another directory when finished.  This continues
#* until the user terminates the program.
#*****************************************************************************

tape_device
logical_tape
directory
rewind_tape

echo " "
echo "Do you want to tar another directory? [y/n]"

while read again
do
  if [ "${again}" = "n" ] || [ "${again}" = "no" ]
  then
    exit
  fi

  logical_tape
  directory
  rewind_tape

  echo " "
  echo "Do you want to tar another directory? [y/n]"
done
```

APPENDIX C

Program to restore Gulf Coast RASA Study Data from a WORM Disk
Written by Kimberly A. Kirkpatrick

```
#! /bin/sh
# Script Name:  Archive Restore
# Author:  Kimberly Kirkpatrick
# Purpose:  To allow a general user to restore data from the Gulf Coast
#   RASA Study
# Date:  September 25, 1992
# Modified:  By Richard Hollway, 9/29/92


#*************************************************************************
#* This procedure tells the user the contents of the WORM disk from the
#* Gulf Coast RASA Study.
#*************************************************************************
header()
{
echo " "
echo "                              Gulf Coast RASA Data "
echo " "
echo "          Logical Tape          Name of Directory          Size  "
echo "          ------------          -----------------          ------ "
echo " "
echo "                1                directory                273 blocks"
echo "                2                cores                    59432 blocks"
echo "                3                kuiper                   11051 blocks"
echo "                4                logs                     7022 blocks"
echo "                5                misc                     105592 blocks"
echo "                6                pisub5                   266 blocks"
echo "                7                qw                       182145 blocks"
echo "                8                rasadata                 175291 blocks"
echo "                9                watlevels                108626 blocks"
echo "                10               wgcrasa                  44565 blocks"
echo "                11               williamson               212 blocks"
}

#*************************************************************************
#* This procedure assigns /dev/rmt/1 as the default tape drive.  It prompts
#* the user to enter another name if the default is not suitable.  It then
#* prompts to see if the tape is mounted and ready before continuing.
#*************************************************************************
tape_device()
{
name="/dev/rmt/1"
echo " "
echo " "
echo "                Gulf Coast RASA Data "
echo " "
echo "This program allows the restoration of Gulf Coast RASA Study data"
```

APPENDIX C--Continued

Program to restore Gulf Coast RASA Study Data from a WORM Disk

```
echo "from the write-once, read-many disk."

while :
do
  echo " "
  echo "Mount the tape and type in the name of the tape device."
  echo "For example:  /dev/pmt/0  (Default: /dev/rmt/1)"
  read namet

  if [ "$namet" != "" ]
  then
    name="$namet"
  fi

  if test -c "$name"
  then
    break
  else
    echo " "
    echo "$name does not exist.  Please check, and try again."
  fi
done

while :
do
  echo " "
  echo "Is the tape mounted and ready [y/n]?"
  read answer

  if [ "${answer}" = "y" ] || [ "${answer}" = "yes" ]
  then
    mt -f ${name}n rew
    break
  else
        echo "Please mount the tape."
        echo ""
  fi
done
}

#*************************************************************************
#* This procedure prompts the user to enter a logical tape number and file
#* steps forward the appropriate number.
#*************************************************************************
logical_tape()
{
echo " "
echo "Which logical tape would you like to restore?  (1-11)"
```

APPENDIX C--Continued

Program to restore Gulf Coast RASA Study Data from a WORM Disk

```
read tape
mt -f ${name}n fsf ${tape}
mt -f ${name}n bsf
}

#*************************************************************************
#* This procedure prompts the user to enter the full pathname of the
#* directory to be restored.  A check is done to make sure the specified
#* pathname is not a file, that it exists, and the the user has write
#* privileges for the directory.  If so, the logical tape is restored in
#* the directory.
#*************************************************************************
directory()
{
while :
do
  echo " "
  echo "Which directory would you like to restore to?  You must specify the"
  echo "full pathname.  (Example:  /rasadata/archive )"
  read dir

  if test -f "$dir"
  then
    echo " "
    echo "$dir is a file, not a directory"
  elif test -w "$dir"
  then
    cd ${dir}
    tar xvf ${name}n
    break
  else
    echo " "

    echo "You do not have the necessary privileges for the $dir directory,"
    echo "or it does not exist.  Please check, and try again."
  fi
done
}

#*************************************************************************
#* This is the main body of the program.  The user is prompted for the tape
#* device, the logical tape number, the full pathname of the directory, and
#* if they would like to restore another logical tape when finished.  This
#* continues until the user terminates the program.
#*************************************************************************

tape_device
header
```

APPENDIX C--Continued

Program to restore Gulf Coast RASA Study Data from a WORM Disk

```
logical_tape
directory
mt -f ${name}n rew

echo " "
echo "Do you want to restore another logical tape [y/n]?"

while read again
do
 if [ "${again}" = "n" ] || [ "${again}" = "no" ]
 then
  exit
 fi
  header
  logical_tape
  directory
  mt -f ${name}n rew

  echo " "
  echo "Do you want to restore another logical tape [y/n]?"
done
```