

UNITED STATES DEPARTMENT OF THE INTERIOR

GEOLOGICAL SURVEY

Multics STATPAC user handbook--Part 2
a guide with examples to basic statistical
programs and more advanced general operation

By

Wendy S. Speckman

Open-File Report 83-737

1983

This report is preliminary and has not been reviewed for
conformity with U.S. Geological Survey editorial standards
and stratigraphic nomenclature.

CONTENTS

Introduction.....	1
Archive Command.....	2
Absentee Requests.....	5
d0010 Fisher-K Statistics.....	10
d0101 Correlation Analysis.....	12
logs10.....	13
creates.....	14
d0101.....	15
d0036 Graphical Analysis.....	16
boundry.....	17
d0036.....	18
Publst program.....	20
References cited.....	23

Introduction

This handbook is designed to be used by persons who have previously used the "Multics STATPAC user handbook--Part 1 A guide with examples for editing and correcting a STATPAC dataset. Much computer 'jargon' is used and most of the text will not be understood by someone unfamiliar with computer terminology and operations.

The structure of the handbook is progressive -- it is intended to be followed from beginning to end. Each new topic builds on information previously explained or referenced. The examples are from actual computer runs.

This handbook discusses and shows examples of the system archive command. This command allows you to combine segments to save space, but each segment retains its integrity. The process used to set up an absentee request is described and several examples are shown. This handbook describes the Fisher-K statistics program ('d0010'), the graphical analysis program with histograms ('d0036'), the correlation analysis program ('d0101'), and the program to obtain a CTR (Central Technical Reports) publication listing ('publst').

The STATPAC system and its supporting programs allow for many kinds of statistical procedures. It is possible to make contour plots and plots based on the concentrations of elements. Other programs available are percentile routines, regression analyses, analysis of variance, ermode and qmode factor analysis.

Throughout this handbook the parts that are underlined are the things the user types. The notation "↵" is used to indicate a carriage return.

'Archive' Command

The 'archive' command allows you to combine segments (usually segments that are related in some way) into an archived segment. The archived segment normally takes up fewer Multics pages than the individual segments and is only one item in your listing of files. Each segment retains its name and integrity in the archived file.

The 'archive' command has many keys (operations). These keys specify the archive operation to be performed. Some of the most commonly used keys are:

t	prints table of contents of archive
a	adds copy of segment to archive
d	deletes segment from an archive
x	extracts copy of segment from archive and puts it in working directory
ad	adds segment to archive and deletes it from working directory

An archived segment must have the suffix '.archive'. You do not need to type the '.archive' when referring to the archived segment through the 'archive' command, but you do need to use the complete name of the file (including the suffix '.archive') when using other Multics commands such as 'cp' and 'dl'.

Examples of the above keys are used in the 'archive' commands on the following two pages.

If all segments to be archived have been named so they are accessible using the wild card .* the active function 'segs' can be used to add all the segments at one time without typing each one individually. The example shows five segments all having the form denver.*, so the active function is used to create the archive having those five segments in it. The 't' key is used to list the contents of the archive. The 'x' key is used to extract one file from the archive. Note that a copy of the segment remains untouched in the archive.

We chose to rename the file "denver.corr4" to "denver.corrected" and decided that the files "denver.corr4", "denver.corr3", "denver.corr2", and "denv.corr1" were not necessary, so they were deleted from the archive "denv.archive," and the file "denver.corrected" was added to the archive, but not deleted from the working directory.

ls

Segments = 5, Lengths = 15.

r w 3 denver.corr4
r w 3 denver.original
r w 3 denver.corr3
r w 3 denver.corr2
r w 3 denver.corr1

r 07/07/83 1306.6 mdt Thu \$0.01 \$0.40

ac ad denv [segs denver.**]

archive: Creating >udd>Cmptappl>WSpeckman>test>denv.archive

r 07/07/83 1306.7 mdt Thu \$0.18 \$0.58

ls

Segments = 1, Lengths = 13.

r w 13 denv.archive

r 07/07/83 1306.7 mdt Thu \$0.00 \$0.59

ac t denv

>udd>Cmptappl>WSpeckman>test>denv.archive

updated		name
07/07/83	1306.7	denver.corr1
07/07/83	1306.7	denver.corr2
07/07/83	1306.7	denver.corr3
07/07/83	1306.7	denver.original
07/07/83	1306.7	denver.corr4

r 07/07/83 1306.7 mdt Thu \$0.01 \$0.60

ac x denv denver.corr4

r 07/07/83 1306.7 mdt Thu \$0.02 \$0.62

ls

Segments = 2, Lengths = 16.

r w 3 denver.corr4
r w 13 denv.archive

r 07/07/83 1306.7 mdt Thu \$0.00 \$0.62

rn denver.corr4 denver.corrected
r 07/07/83 1306.7 mdt Thu \$0.05 \$0.67

ac d denv denver.corr4 denver.corr3 denver.corr2 denver.corr1
r 07/07/83 1306.7 mdt Thu \$0.02 \$0.69

ac a denv denver.corrected
r 07/07/83 1306.7 mdt Thu \$0.05 \$0.73

ls

Segments = 2, Lengths = 9.

r w 3 denver.corrected
r w 6 denv.archive

r 07/07/83 1306.7 mdt Thu \$0.00 \$0.74

ac t denv

>udd>Cmptappl>WSpeckman>test>denv.archive

updated		name
07/07/83	1306.7	denver.original
07/07/83	1306.7	denver.corrected
r 07/07/83	1306.7	mdt Thu \$0.01 \$0.74

Absentee Requests

Any procedure that you do interactively (without operator intervention) can be done as an absentee process. The main reason for using an absentee request is to save money. Absentee requests are normally processed when the costs are lower (in queue 3 or queue 4). To set up an absentee file you need to use a text editor; 'taco' has been used in the examples, but 'qedx' could be used. The file that controls the absentee process needs to have the suffix '.absin'. The request to execute an absentee process is entered by using the 'ear' (enter_abs_request) command. When the job is processed by the system a file having a suffix of '.absout' is created. This file should be examined to see if the procedure executed properly.

The information put into the '.absin' file must contain all the commands and responses you would type if you were doing the procedure interactively.

The first example (see following page) is an absentee process to execute 'd0039' to get a publication listing. Before setting up the '.absin' file for any STATPAC program suffixed by 'A', 'B', or 'D' you need to create the necessary program control file. It is normally easier to create this file interactively, but the program control file can be created as the first step in the absentee process. Once that file has been created interactively, you need to use a text editor to create the '.absin' file.

The first line in the '.absin' file should be a 'cwd' command to move your process to the proper working directory. (This is necessary only if you have subdirectories; if you work directly under your person_id this line is not needed.) The remaining lines in the file invoke the program and give the required responses to the questions asked by the program. You need to know what the questions will be for whatever program you try to execute. You can 'rename' and 'dprint' files in the absentee process. The last line in every '.absin' file should be a 'logout' command. If you make typing errors use the text editor to correct them.

Once the '.absin' file is correct you can enter the request to the system to execute the job. You can tell the system to execute the job at a given time or simply in which 'queue'. This example asks the job to be entered in queue 4. The system will process the request as soon as it can, and you will be charged queue 4 rates rather than the queue 1 rates you would have been charged had you executed the program interactively. It is ALWAYS advisable to enter absentee requests in queue 4.

The next time you 'login', you need to 'list' your segments. There will always be a file suffixed with '.absout' -- even if the procedure did not do what you wanted. In this example another file was created and also is shown on the list. You can 'print' the '.absout' file to see if the procedure executed to completion. In this example you can see that the '.absout' file looks like a good interactive session execution of 'd0039'. If you omit any answer from your '.absin' file that the program expects, the procedure will not run to completion and you will have to either fix the '.absin' file and re-enter the request or complete the process interactively.

Delete the '.absin' and '.absout' files once you have checked the process.

The second absentee example demonstrates how to get a listing on the line printer from 'lookst' of tag number, field number, latitude, and longitude. Every response that 'lookst' expects has to be in the '.absin' file. The request is entered with the 'ear' command.

When you do a 'ls' command in the next login session you will see both the '.absin' and the '.absout' files. Do not print the '.absout' file in this case because it has all the information from 'lookst' in it and is quite long. The 'dp look.absout' line in the '.absin' file 'dprinted' the '.absout' file and you can check it completely when you receive the printout.

head39

What is the input file name? : denver.corrected

Enter output card file name : d39

title for listing (30 char) : denver sediments -corrected

Do you want row numbers printed? : y

Do you want lat.-long. in deg.,min.,sec. ? : y

STOP

r 07/07/83 1306.7 mdt Thu \$0.09 \$0.83

teco d39.absin

teco: Entry not found. d39.absin

@!go >udd>Cmptappl>WSpeckman>test

exec d0039 y

no

d39

denver.corrected

no

rn d0039.*.list d0039.denver.corrected

dp d0039.denver.corrected

logout

!\$

@mweq\$

r 07/07/83 1306.8 mdt Thu \$0.15 \$0.98

pr d39.absin 1

go >udd>Cmptappl>WSpeckman>test

exec d0039 y

no

d39

denver.corrected

no

rn d0039.*.list d0039.denver.corrected

dp d0039.denver.corrected

ls

Segments = 4, Lengths = 11.

r w 1 d39.absin

r w 1 d39

r w 3 denver.corrected

r w 6 denv.archive

r 07/07/83 1306.8 mdt Thu \$0.01 \$1.02

ear d39 -q 4

ID: 190649.3; 21 already requested.

r 07/07/83 1306.8 mdt Thu \$0.05 \$1.06

ls

Segments = 6, Lengths = 20.

```
r w    8  d0039.denver.corrected
r w    1  d39.absout
r w    1  d39.absin
r w    1  d39
r w    3  denver.corrected
r w    6  denv.archive
```

```
r 07/07/83 1328.1 mdt Thu $0.01 $0.25
```

pr d39.absout 1

```
Absentee user wSpeckman Cmptappl logged in: 07/07/83 1326.7 mdt Thu
r 07/07/83 1326.9 mdt Thu $0.23 $0.23
```

```
go >udd>Cmptappl>WSpeckman>test
r 07/07/83 1326.9 mdt Thu $0.01 $0.24
```

```
exec d0039 y
```

```
Do you want printer output on TTY      ?    no
```

```
What is program control file name      ?    d39
```

```
What is input binary file name         ?    denver.corrected
```

```
STOP
```

```
Do you want your printer file queued ?    no
r 07/07/83 1327.2 mdt Thu $0.67 $0.91
```

```
rn d0039.*.list d0039.denver.corrected
r 07/07/83 1327.2 mdt Thu $0.01 $0.92
```

```
dp d0039.denver.corrected
1 request signalled, 0 already in printer queue 4
r 07/07/83 1327.2 mdt Thu $0.03 $0.95
```

```
logout
```

```
r $0.97
r 07/07/83 1328.1 mdt Thu $0.01 $0.26
```

dl d39.absin d39.absout d39

```
r 07/07/83 1328.1 mdt Thu $0.01 $0.27
```

```

teco look.absin
teco: Entry not found. look.absin
@i!go >udd>Cmptappl>WSpeckman>test
lookst
denver.corrected
0
y
y
n
y
0
y
stop
dp look.absout
logout
!$
@0jht$
go >udd>Cmptappl>WSpeckman>test
lookst
denver.corrected
0
y
y
n
y
0
y
stop
dp look.absout
logout
@mweq$
r 07/07/83 1328.2 mdt Thu $0.08 $0.35

```

```

ear look -q 4
ID: 192812.5; 21 already requested.
r 07/07/83 1328.2 mdt Thu $0.03 $0.38

```

```

ls look.**

Segments = 2, Lengths = 3.

r w      2 look.absout
rew     1 look.absin

r 07/07/83 1340.6 mdt Thu $0.01 $0.26

```

```

dl look.**
r 07/07/83 1340.6 mdt Thu $0.01 $0.27

```

'd0010' Program
Fisher-K Statistics

For 'd0010' (as for all STATPAC programs prefixed by 'A', 'B', or 'D') you need a program control card file. The control file needed for 'd0010' is normally very simple because most people use the default options. If you want to use options other than the default options you will need to run the program 'creates' to set up the program control file. To get a copy of the documentation for 'd0010' type:

```
dp >uml>statpac>doc>d0010.runout
```

If you use the defaults for 'd0010' you can use the program 'header' to create this file. The only information contained in the program control card file in this case is the dataset_id and number of rows and columns in the dataset.

The following page is an example of 'header', the execution of 'd0010', and the renaming and dprinting of the printer file.

'd0101' Program
Correlation Analysis

Correlation analysis can be run on a file with raw data or log data. To transform raw data to log data execute the program 'logs10'. This creates a new file in which all data have been changed to logs.

To make the program control card file for 'd0101' use the program 'creates'. The first prompt is for the name of the file (program control card file) you are creating. The next prompt asks for the program you wish to execute. The options used in the following example are common ones. The documentation for 'd0101' explains all possible options. To get a copy of the documentation type:

```
dp >uml>statpac>doc>d0101.runout
```

If you want to eliminate coordinates from the correlation matrix in this example, you would select a column pair of 3,36 rather than 1,36.

When you execute 'd0101' you can use the raw data file as input or the log data file as input -- as long as both of those files have the same dataset_id and number of rows and columns.

logs10

Enter input binary file : denver.corrected

Enter output binary file : denver.logs

Enter D.S. name (or c.r. to exit) : segs

Do you want selected columns ? no

Data Set segs terminated normally.

Enter D.S. name (or c.r. to exit) : ↵

STOP

r 07/07/83 1340.8 mdt Thu \$0.11 \$1.07

creates

Welcome to the USGS program control card builder for "STATPAC" Programs
Currently this program will create cards for 35 of the programs.

Enter Control File Name = d101
Enter Stapac Prog to be run = d0101
Enter Title (30 chars) = denver.sediments
Enter Input Data Set ID = segs
Enter Input Number of Rows = 50
Enter Input Number of Cols = 36

Enter Options :

Option(1) = 1
Option(2) = 0
Option(3) = 1
Option(4) = 1

Enter Output Data Set ID = segs
Enter Output Number of Rows = 50
Enter Output Number of Cols = 36
Enter No. of Row Pairs = 1
Enter No. of Column Pairs = 1

denver.sediments segs 50 361011000000segs

50 3

OK ? y

COLUMN SELECTOR CARDS :

From-To (xxx,xxx) ? 1,36
1 36

OK ? y

ROW SELECTOR CARDS :

From-To (xxx,xxx) ? 1,50
1 50

OK ? y

USE TECO TO ADD OR CHANGE ANY CARDS TO THIS FILE.

ANOTHER SET OF DATA FOR THE SAME PROGRAM ?

no

ANOTHER SET OF DATA FOR A DIFFERENT PROGRAM ?

no

STOP

r 07/07/83 1340.9 mdt Thu \$0.10 \$1.16

exec d0101 y

Do you want printer output on TTY ? no
what is program control file name ? d101
What is input binary file name ? denver.corrected

STOP

Do you want your printer file queued ? no
r 07/07/83 1341.2 mdt Thu \$1.09 \$2.26

ls -ft 2

Segments = 8, Lengths = 38.

r w 18 d0101.!BBBBJNBKhzwqdL.list
r w 1 d101
r 07/07/83 1341.2 mdt Thu \$0.01 \$2.26
rn d0101.*.list d101.denver.corrected
r 07/07/83 1341.2 mdt Thu \$0.01 \$2.27

exec d0101 y

Do you want printer output on TTY ? no
what is program control file name ? d101
what is input binary file name ? denver.logs

STOP

Do you want your printer file queued ? no
r 07/07/83 1341.6 mdt Thu \$1.08 \$3.35

ls -ft 2

Segments = 9, Lengths = 56.

r w 18 d0101.!BBBBJNBKjBDHbkl.list
r w 18 d101.denver.corrected
r 07/07/83 1341.6 mdt Thu \$0.01 \$3.35
rn d0101.*.list d101.denver.corrected.logs
r 07/07/83 1341.6 mdt Thu \$0.01 \$3.36

dp d101.denver.corrected d101.denver.corrected.logs
2 requests signalled, 0 already in printer queue 4
r 07/07/83 1341.6 mdt Thu \$0.02 \$3.38

'd0036' Program
Graphical Analysis

'd0036' calculates the univariate and bivariate frequency distribution of a dataset. It uses graphical displays to show histograms and contingency tables.

To create the program control card file for 'd0036' you need to run the program 'boundry'. This program automatically puts in the lower boundary and class interval commonly used for spectrographic data (logs) for each variable in the dataset. The example of 'boundry' shows options and percentiles commonly used. The options are explained in the documentation for 'd0036'. To get a copy of the documentation type:

```
dp >uml>statpac>doc>d0036.runout
```

The listing of cards created is a printout of the program control card file created by 'boundry'.

After you create the program control file you need to execute the program 'd0036', 'rename' the printer file, and 'dprint' the printer file.

NOTE: If you wish you can go through a routine to replace the qualified values with actual values so they are reflected in the histogram. The routines that you can use to do this are (1) the 'replac' command in 'genstat' or (2) the stand-alone program 'replac'. The stand-alone program 'replac' is one of the programs written by Al Miesch. The documentation for all of his programs can be obtained by typing:

```
dp >uml>statpac>doc>atm.abstracts
```

boundary

This program reads a STATPAC binary file and creates a control file ready for input to Graphical Analysis (a0036) or Geosum (A470). The card file will contain the header record, lower boundary, & class interval for each column for each DS in the file.

Enter Statpac file : denver.logs

Enter Output file : a36

Is data in logs ? yes

Does cols 1 & 2 contain coordinates ? yes

Enter title (30 chars max) : denver.sediments

Enter D0036 or A470 Program Options :

- Option(1) = 1
- Option(2) = 0
- Option(3) = 0
- Option(5) = 2
- Option(6) = 1
- Option(8) = 0
- Option(9) = 0

Selected percentiles (y or n) ? y

Terminate entry with 0.0; 10 percentiles max allowed.

- 1th per = 25
- 2th per = 50
- 3th per = 75
- 4th per = 90
- 5th per = 95
- 6th per = 98
- 7th per = 99
- 8th per = 0

Do want to see a listing of the cards created for the 1th file ? yes

```
denver.sediments          seds          50 361000210000
 3 36
 0.083 -0.417 -0.250 -0.584 2.4169999.999 2.2509999.999 0.91
 0.083 0.916 1.250 0.750 0.916 0.750 1.416 0.583 0.91
 1.4169999.999 0.750 0.916 2.083 1.5839999.999 1.250 2.25
9999.999 0.416 0.583 0.250
0.1666670.1666670.1666670.1666670.1666679.9999900.1666679.9999900.16666
0.1666670.1666670.1666670.1666670.1666670.1666670.1666670.1666670.16666
0.1666679.9999900.1666670.1666670.1666670.1666679.9999900.1666670.16666
9.9999900.1666670.1666670.166667
 25.0 50.0 75.0 90.0 95.0 98.0 99.0 0.0 0.0 0.0
```

STOP

r 07/08/83 1525.3 mdt Fri \$0.19 \$0.45

exec d0036 y

Do you want printer output on TTY ? no
What is program control file name ? d36
What is input binary file name ? denver.logs

STOP

Do you want your printer file queued ? no
r 07/08/83 1457.3 mdt Fri \$0.55 \$1.03

ls -ft 2

Segments = 11, Lengths = 81.

r w 20 d0036.!BBBJNBNNjcQCwB.list
r w 1 d36
r 07/08/83 1457.3 mdt Fri \$0.01 \$1.04

rn d0036.*.list d36.denver.logs
r 07/08/83 1457.3 mdt Fri \$0.01 \$1.04

dp d36.denver.logs

1 request signalled, 0 already in printer queue 4
r 07/08/83 1457.4 mdt Fri \$0.05 \$1.09

ls

Segments = 11, Lengths = 81.

r w 20 d36.denver.logs
r w 1 d36
r w 18 d101.denver.corrected.logs
r w 18 d101.denver.corrected
r w 1 d101
r w 3 denver.logs
r w 3 d10.denver.corrected
r w 1 d10
r w 8 d0039.denver.corrected
r w 3 denver.corrected
r w 8 denv.archive
r 07/08/83 1457.4 mdt Fri \$0.01 \$1.10

dl d36.denver.logs d36 d101.denver.corrected.logs d101.denver.corrected d101
r 07/08/83 1457.4 mdt Fri \$0.02 \$1.12

dl d10.denver.corrected d10 d0039.denver.corrected
r 07/08/83 1457.4 mdt Fri \$0.01 \$1.13

ls

Segments = 3, Lengths = 14.

r w 3 denver.logs
r w 3 denver.corrected
r w 8 denv.archive

r 07/08/83 1457.4 mdt Fri \$0.00 \$1.14

ac ad denv denver.logs
r 07/08/83 1457.5 mdt Fri \$0.03 \$1.16

ac t denv

>udd>Cmptappl>WSpeckman>test>denv.archive

updated		name
07/07/83	1306.7	denver.original
07/07/83	1306.7	denver.corrected
07/07/83	1341.8	denver.logs

r 07/08/83 1457.5 mdt Fri \$0.01 \$1.17

dl denver.corrected
r 07/08/83 1457.5 mdt Fri \$0.01 \$1.17

ls

Segments = 1, Lengths = 8.

r w 8 denv.archive

r 07/08/83 1457.5 mdt Fri \$0.00 \$1.18

'publst' Program

To obtain a publication listing in CTR format use the program 'publst'. This program eliminates trailing zeros in data. It puts in '--' instead of 0.0B, < instead of L, and N instead of N and the value. The program can use TRU standard column_ids instead of the all upper-case variable_ids you see in the other STATPAC programs. This program can be executed without a control deck if the title to be used is only one line. If you want sub-titles, you need to make up a control deck. This is explained further in the documentation on 'publst'. To get a copy of this documentation type:

```
dp >uml>statpac>doc>publst.info
```

The following example doesn't use a control deck. I chose TRU format 1, TRU standard column_ids, and 5 lines per group.

NOTE: If TRU standard column_ids are used, the 'dp' command MUST have a special control argument -- '-pl 61'. This control argument changes the line printer from 60 to 61 lines per page. This is necessary because the TRU standard column_ids are two lines. If you neglect to 'dp' with this control argument, your printout will be missing a sample at the bottom of each page.

publst

Publication listing program PUBLST(Nov 1982 version)

See documentation for operation revisions (pathname below)

>uml>statpac>doc>helpub

There are 2 changes affecting the user in this version:

- (1) A default mode in which only 3 inputs are required.
(Type "default" when prompted for control deck file)
- (2) If dataset id = "icp", the output response is different for censored data containing "N" or "L" codes. See the documentation for further details.

enter input filename(to be listed): denver.corrected

enter filename of control deck: ↓

enter filename of output(lpt) file: List-denver-seds

enter dataset id (or "q","exit" to stop): seds

enter page width (0 => output on printer): 0

select format (1=tru1,2=tru2,3=d0039): 1

type (y) if any other options are desired: y
[answer the following questions (except 3) with y or n:]

geodetic coordinates in deg,min & secs?(y/n)y

type (y) to use tru standard column ids: y
Line printer selected. When dprinting the output file:
Set pl option value to 61. ex: dp -pl 61 output.file

enter number of lines/group (if not =5): 5

		output col.	ids		
Latitude	Longitude	Fe-pct.	Mg-pct.	Ca-pct.	Ti-pct.
		s	s	s	s
Mn-ppm	Ag-ppm	As-ppm	Au-ppm	B-ppm	Ba-ppm
s	s	s	s	s	s
Be-ppm	Bi-ppm	Cd-ppm	Co-ppm	Cr-ppm	Cu-ppm
s	s	s	s	s	s
La-ppm	Mo-ppm	Nb-ppm	Ni-ppm	Pb-ppm	Sb-ppm
s	s	s	s	s	s
Sc-ppm	Sn-ppm	Sr-ppm	V-ppm	W-ppm	Y-ppm
s	s	s	s	s	s
Zn-ppm	Zr-ppm	Th-ppm	Cu-ppm	Pb-ppm	Zn-ppm
s	s	s	aa	aa	aa

if any editing needs to be done, type (y) n

enter main title:

Denver sediments

For dataset [seds] there are 50 X 36 rows and columns.

enter dataset id (or "q","exit" to stop): exit

STOP

r 07/07/83 1344.6 mdt Thu \$1.56 \$2.19

dp -pl 61 list-denver-seds

1 request signalled, 0 already in printer queue 4

r 07/07/83 1344.6 mdt Thu \$0.05 \$2.24

References Cited

U.S.G.S. Multics Users Manual, Computer Center Division,
June, 1978

Vantrump, George, Jr., and Miesch, A. T., 1977, The U.S.
Geological Survey RASS-STATPAC System for Management and
Statistical reduction of geochemical data: Computers
and Geosciences v. 3, p. 475-488.