Data 520 Introduction to Programming The 9522: A python project

Heidi Beezub

December 6, 2017

Figure: Beezub

Outline

The 9522 and Rejects

- The DCM 9522 is one of the reports I use in order to correct 'rejects' in my job. DCM stands for Department Cost Manager. It includes multiple departments and all the products (both supplies and services) that each department provides.

The 9522 and Rejects

- The DCM 9522 is one of the reports I use in order to correct 'rejects' in my job. DCM stands for Department Cost Manager. It includes multiple departments and all the products (both supplies and services) that each department provides.
- **What is a reject?**
  A reject is data that the system collects but then cannot process or put into the correct category. Rejects are typically a 'new' item that simply needs the interface created or built in the system. Sometimes this is an existing item that is coming into the system in a new way and needs an additinal interface.

The 9522 and Rejects

- The DCM 9522 is one of the reports I use in order to correct 'rejects' in my job. DCM stands for Department Cost Manager. It includes multiple departments and all the products (both supplies and services) that each department provides.
- **What is a reject?**
  A reject is data that the system collects but then cannot process or put into the correct category. Rejects are typically a 'new' item that simply needs the interface created or built in the system. Sometimes this is an existing item that is coming into the system in a new way and needs an additinal interface.
- **What is an interface?** The interface matches the data into the correct department. The system matches the product information (feeder key and feeder system).

9522 could be better

- The 9522 helps me to determine if this is an existing item for which I need to create an additional interface. Or if it is a new product, which means I need to create the new product (including the interface).

9522 could be better

- The 9522 helps me to determine if this is an existing item for which I need to create an additional interface. Or if it is a new product, which means I need to create the new product (including the interface).
- The report is **unwieldy** (6,600 pages). I have to save the text file as a word file and then search the word file for the feeder key, IP number, or description information. Since this is typically numerical it can result in multiple 'hits' many for the wrong field. I also cannot easily compare the multiple hits in the word file.

9522 could be better

- The 9522 helps me to determine if this is an existing item for which I need to create an additional interface. Or if it is a new product, which means I need to create the new product (including the interface).
- The report is **unwieldy** (6,600 pages). I have to save the text file as a word file and then search the word file for the feeder key, IP number, or description information. Since this is typically numerical it can result in multiple 'hits' many for the wrong field. I also cannot easily compare the multiple hits in the word file.
- As an excel file I would be able to search on the columns and view any similar items (all at the same time). It helps in identifying items to see what department(s) similar items are used in.

Methodology

- **Source data** My first step was to take a VERY critical look at the structure of the report. The report includes headers on each page which can bisect the data at any point. In addition to the information I want, the data also includes data I don't want that is part of the Bill of Materials.

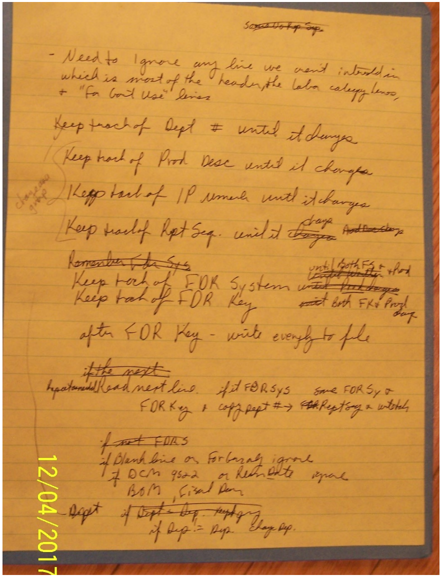
Methodology

- **Source data** My first step was to take a VERY critical look at the structure of the report. The report includes headers on each page which can bisect the data at any point. In addition to the information I want, the data also includes data I don't want that is part of the Bill of Materials.
- **The desired end result** A csv (excel file) that I will be able to filter to check items, sort to identify duplicate products and otherwise manipulate the data. With the goal being a better, more accurate and consistent database.

Methodology

- **Source data** My first step was to take a VERY critical look at the structure of the report. The report includes headers on each page which can bisect the data at any point. In addition to the information I want, the data also includes data I don't want that is part of the Bill of Materials.
- **The desired end result** A csv (excel file) that I will be able to filter to check items, sort to identify duplicate products and otherwise manipulate the data. With the goal being a better, more accurate and consistent database.
- **Pseudocode** First, I wrote down (pen paper) what I needed to do.

Pseudocode



~~[illegible]~~

- Need to Ignore any line we aren't interested in which is most of the header, the labor category lines, + "For Gov't Use" lines

Keep track of Dept # until it changes

changes group

Keep track of Prod Desc until it changes

Keep track of IP number until it changes

Keep track of Rpt Seq. until it ~~changes~~ changes ~~[illegible]~~

~~Remember [illegible] Sys~~

Keep track of FOR System ~~until [illegible] changes~~ until Both FS + ~~[illegible]~~ + Prod

Keep track of FOR Key ~~until~~ Both FK + Prod change

after FOR Key - write everything to file

~~if the next~~

~~[illegible]~~ Read next line. if it FOR Sys save FOR Sys &
FOR Key + copy Dept # → ~~[illegible]~~ RptSeq & [illegible]

if ~~not FORS~~

if Blank line or Forbearing ignore

if DCM 9522 or Run Date ignore

BOM, Fiscal Year

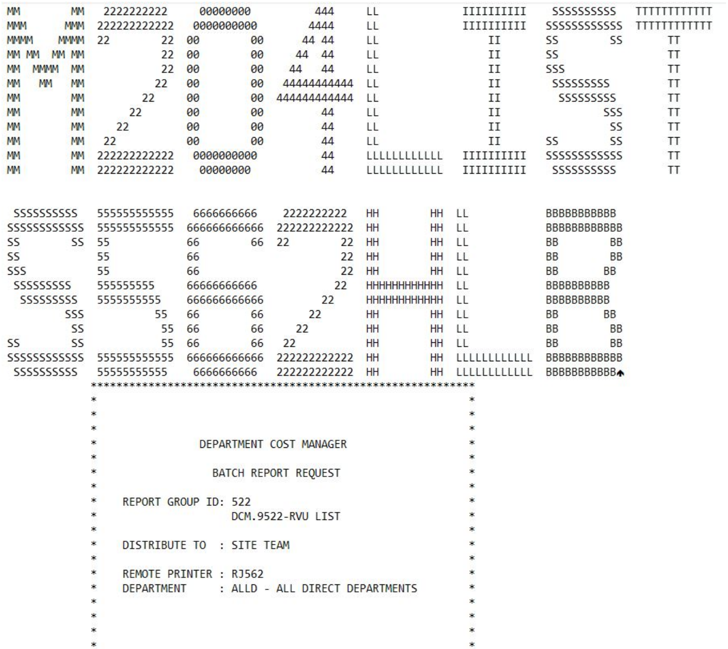
~~Dept~~ if ~~Dept = Dep. [illegible]~~

if Dep != Dep. Change Dep.

12/04/2017

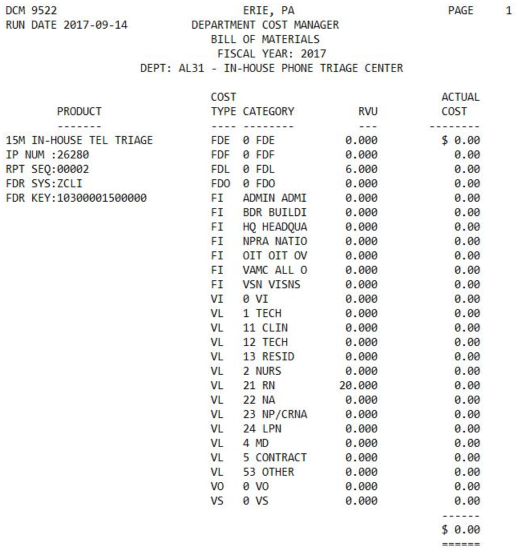
Figure: 9522 pseudocode

9522 pre



```
MM        MM   2222222222     00000000          444     LL             IIIIIIIIII    SSSSSSSSSS   TTTTTTTTTTTT
MMM      MMM  222222222222   0000000000        4444     LL             IIIIIIIIII   SSSSSSSSSSSS  TTTTTTTTTTTT
MMMM    MMMM  22        22  00        00      44 44     LL                 II       SS        SS       TT
MM MM  MM MM            22  00        00     44  44     LL                 II       SS                 TT
MM  MMMM  MM            22  00        00    44   44     LL                 II       SSS                TT
MM   MM   MM           22   00        00   44444444444  LL                 II        SSSSSSSSS         TT
MM        MM         22     00        00  444444444444  LL                 II         SSSSSSSSS        TT
MM        MM       22       00        00         44     LL                 II                SSS       TT
MM        MM     22         00        00         44     LL                 II                 SS       TT
MM        MM   22           00        00         44     LL                 II       SS        SS       TT
MM        MM  222222222222   0000000000          44     LLLLLLLLLLLL  IIIIIIIIII   SSSSSSSSSSSS       TT
MM        MM  222222222222    00000000           44     LLLLLLLLLLLL  IIIIIIIIII    SSSSSSSSSS        TT


 SSSSSSSSSS   555555555555   6666666666    2222222222   HH        HH  LL            BBBBBBBBBBB
SSSSSSSSSSSS  555555555555  666666666666  222222222222  HH        HH  LL            BBBBBBBBBBBB
SS        SS  55            66        66  22        22  HH        HH  LL            BB        BB
SS            55            66                      22  HH        HH  LL            BB        BB
SSS           55            66                      22  HH        HH  LL            BB       BB
 SSSSSSSSS    555555555     66666666666            22   HHHHHHHHHHHH  LL            BBBBBBBBBB
  SSSSSSSSS   5555555555    666666666666         22     HHHHHHHHHHHH  LL            BBBBBBBBBB
         SSS           55   66        66       22       HH        HH  LL            BB       BB
          SS            55  66        66     22         HH        HH  LL            BB        BB
SS        SS            55  66        66   22           HH        HH  LL            BB        BB
SSSSSSSSSSSS  555555555555  666666666666  222222222222  HH        HH  LLLLLLLLLLLL  BBBBBBBBBBBB
 SSSSSSSSSS   55555555555    6666666666   222222222222  HH        HH  LLLLLLLLLLLL  BBBBBBBBBBB
             ************************************************************
             *                                                          *
             *                                                          *
             *                                                          *
             *                DEPARTMENT COST MANAGER                   *
             *                                                          *
             *                  BATCH REPORT REQUEST                    *
             *                                                          *
             *    REPORT GROUP ID: 522                                  *
             *                     DCM.9522-RVU LIST                    *
             *                                                          *
             *    DISTRIBUTE TO  : SITE TEAM                            *
             *                                                          *
             *    REMOTE PRINTER : RJ562                                *
             *    DEPARTMENT     : ALLD - ALL DIRECT DEPARTMENTS        *
             *                                                          *
             *                                                          *
             *                                                          *
             *                                                          *
```

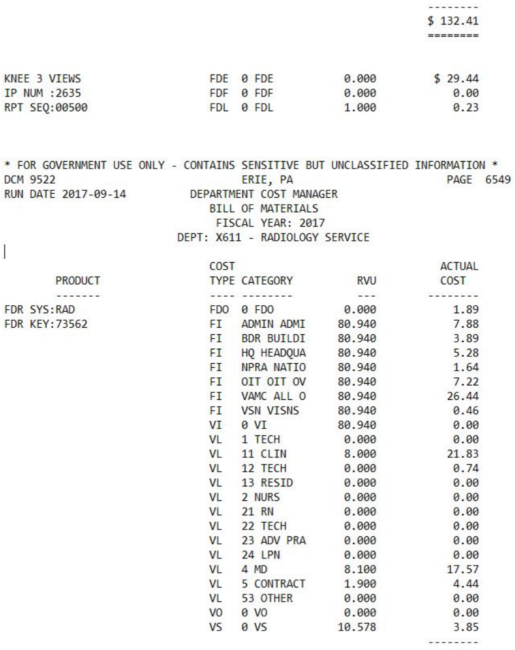
9522 page 1



```
DCM 9522                        ERIE, PA                    PAGE     1
RUN DATE 2017-09-14          DEPARTMENT COST MANAGER
                                BILL OF MATERIALS
                                 FISCAL YEAR: 2017
                      DEPT: AL31 - IN-HOUSE PHONE TRIAGE CENTER

                               COST                             ACTUAL
        PRODUCT                TYPE CATEGORY        RVU         COST
        -------                ---- --------        ---       --------
15M IN-HOUSE TEL TRIAGE        FDE  0 FDE          0.000      $ 0.00
IP NUM :26280                  FDF  0 FDF          0.000        0.00
RPT SEQ:00002                  FDL  0 FDL          6.000        0.00
FDR SYS:ZCLI                   FDO  0 FDO          0.000        0.00
FDR KEY:10300001500000         FI   ADMIN ADMI     0.000        0.00
                               FI   BDR BUILDI     0.000        0.00
                               FI   HQ HEADQUA     0.000        0.00
                               FI   NPRA NATIO     0.000        0.00
                               FI   OIT OIT OV     0.000        0.00
                               FI   VAMC ALL O     0.000        0.00
                               FI   VSN VISNS      0.000        0.00
                               VI   0 VI           0.000        0.00
                               VL   1 TECH         0.000        0.00
                               VL   11 CLIN        0.000        0.00
                               VL   12 TECH        0.000        0.00
                               VL   13 RESID       0.000        0.00
                               VL   2 NURS         0.000        0.00
                               VL   21 RN         20.000        0.00
                               VL   22 NA          0.000        0.00
                               VL   23 NP/CRNA     0.000        0.00
                               VL   24 LPN         0.000        0.00
                               VL   4 MD           0.000        0.00
                               VL   5 CONTRACT     0.000        0.00
                               VL   53 OTHER       0.000        0.00
                               VO   0 VO           0.000        0.00
                               VS   0 VS           0.000        0.00
                                                              ------
                                                              $ 0.00
                                                              ======
```

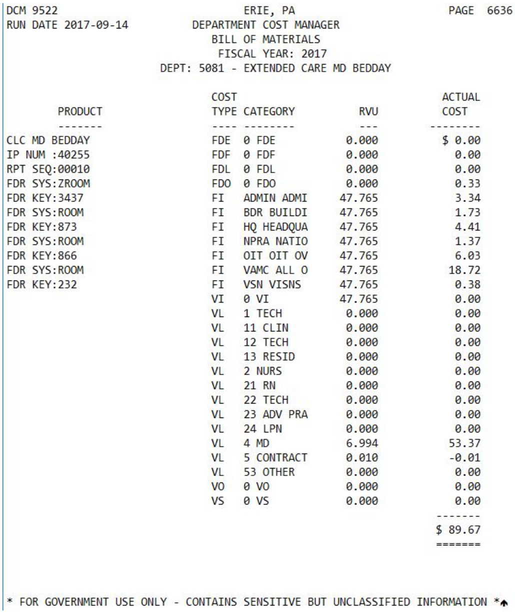
Figure: 9522 page 1

9522 header bisects data



```
                                                                   --------
                                                                   $ 132.41
                                                                   ========


KNEE 3 VIEWS                    FDE  0 FDE          0.000          $ 29.44
IP NUM :2635                    FDF  0 FDF          0.000             0.00
RPT SEQ:00500                   FDL  0 FDL          1.000             0.23



* FOR GOVERNMENT USE ONLY - CONTAINS SENSITIVE BUT UNCLASSIFIED INFORMATION *
DCM 9522                            ERIE, PA                            PAGE  6549
RUN DATE 2017-09-14         DEPARTMENT COST MANAGER
                               BILL OF MATERIALS
                                FISCAL YEAR: 2017
                         DEPT: X611 - RADIOLOGY SERVICE


                                COST                                ACTUAL
        PRODUCT                 TYPE CATEGORY          RVU          COST
        -------                 ---- --------          ---          --------
FDR SYS:RAD                     FDO  0 FDO          0.000             1.89
FDR KEY:73562                   FI   ADMIN ADMI    80.940             7.88
                                FI   BDR BUILDI    80.940             3.89
                                FI   HQ HEADQUA    80.940             5.28
                                FI   NPRA NATIO    80.940             1.64
                                FI   OIT OIT OV    80.940             7.22
                                FI   VAMC ALL O    80.940            26.44
                                FI   VSN VISNS     80.940             0.46
                                VI   0 VI          80.940             0.00
                                VL   1 TECH         0.000             0.00
                                VL   11 CLIN        8.000            21.83
                                VL   12 TECH        0.000             0.74
                                VL   13 RESID       0.000             0.00
                                VL   2 NURS         0.000             0.00
                                VL   21 RN          0.000             0.00
                                VL   22 TECH        0.000             0.00
                                VL   23 ADV PRA     0.000             0.00
                                VL   24 LPN         0.000             0.00
                                VL   4 MD           8.100            17.57
                                VL   5 CONTRACT     1.900             4.44
                                VL   53 OTHER       0.000             0.00
                                VO   0 VO           0.000             0.00
                                VS   0 VS          10.578             3.85
                                                                   --------
```

9522 page 6636



```
DCM 9522                          ERIE, PA                       PAGE  6636
RUN DATE 2017-09-14          DEPARTMENT COST MANAGER
                                BILL OF MATERIALS
                                 FISCAL YEAR: 2017
                        DEPT: 5081 - EXTENDED CARE MD BEDDAY

                                 COST                          ACTUAL
        PRODUCT                  TYPE CATEGORY         RVU      COST
        -------                  ---- --------         ---    --------
CLC MD BEDDAY                    FDE  0 FDE          0.000    $ 0.00
IP NUM :40255                    FDF  0 FDF          0.000      0.00
RPT SEQ:00010                    FDL  0 FDL          0.000      0.00
FDR SYS:ZROOM                    FDO  0 FDO          0.000      0.33
FDR KEY:3437                     FI   ADMIN ADMI    47.765      3.34
FDR SYS:ROOM                     FI   BDR BUILDI    47.765      1.73
FDR KEY:873                      FI   HQ HEADQUA    47.765      4.41
FDR SYS:ROOM                     FI   NPRA NATIO    47.765      1.37
FDR KEY:866                      FI   OIT OIT OV    47.765      6.03
FDR SYS:ROOM                     FI   VAMC ALL O    47.765     18.72
FDR KEY:232                      FI   VSN VISNS     47.765      0.38
                                 VI   0 VI          47.765      0.00
                                 VL   1 TECH         0.000      0.00
                                 VL   11 CLIN        0.000      0.00
                                 VL   12 TECH        0.000      0.00
                                 VL   13 RESID       0.000      0.00
                                 VL   2 NURS         0.000      0.00
                                 VL   21 RN          0.000      0.00
                                 VL   22 TECH        0.000      0.00
                                 VL   23 ADV PRA     0.000      0.00
                                 VL   24 LPN         0.000      0.00
                                 VL   4 MD           6.994     53.37
                                 VL   5 CONTRACT     0.010     -0.01
                                 VL   53 OTHER       0.000      0.00
                                 VO   0 VO           0.000      0.00
                                 VS   0 VS           0.000      0.00
                                                               -------
                                                              $ 89.67
                                                               =======

* FOR GOVERNMENT USE ONLY - CONTAINS SENSITIVE BUT UNCLASSIFIED INFORMATION *
```

Code for IP Number

I wanted to start small and to start in the 'middle' of my code. Each line of text is read into a list and I use the position in the list to extract the data.

```python

def process_file(reader):
    result_line= ''
    result=''
   #first we need to add headers

    with open('9522_new.csv', 'a') as output_file:
        output_file.write('"DEPT","PRODUCT","IPNUM","RPTSEQ","FDRSYS","FDRKEY"' +'\n')
    for line in reader:
        line=line.strip()    #removes leading/trailing whitespace
        field = line.split()
```

Code for IP Number continued

Here is the code I used to extract the IP Number:

```
if len(field)>3:
        for i in range(0,2):
           #find IPNUM
           if field[i] == 'IP' and field[i+1] =='NUM':
                #save IPNUM
                ipnum=field[i+2].strip(':')
```

Code for Sequence Number and Feeder System

I added code one item at a time, this not only made sure I didn't break what I already had, but also allowed me to debug after each step in the process.

Selected code to pull each item:

```python
    #find RPT Seq
     if field[i] == 'RPT' and field[i+1].startswith('SEQ:'):
        #save RPT SEQ
         rptseq =field[i+1].strip('SEQ:')
    #find FDR SYS
     if field[i] == 'FDR' and field[i+1].startswith('SYS:'):
        #(Kudgel) strip SYS deletes the leading/trailing S from the fdrsys
         if field[i+1] == 'SYS:SUR':
            #save FDR SYS
             fdrsys='SUR'
         else:
            #save FDR SYS
             fdrsys=field[i+1].lstrip('SYS:')  #lstrip accounts for ECS fdr system
```

Code for Feeder Key

The **feeder key** is the 'last' item before the line can be written to the file. The majority of Feeder Keys are one 'word'/text string without breaks, however there were a handful of cases where the Feeder Key was more than one text string.

```
    #find FDR KEY (last item before write to file)
if field[i] == 'FDR' and field[i+1].startswith('KEY:'):
        #(Kudgel)save FDR KEY
        if field[i+2] == 'COST':    #for 'MEDIUM COST' fdrkey
            fdrkey=field[i+1].lstrip('KEY:') + ' COST'
        elif field[i+2] == 'DRUG':
            fdrkey=field[i+1].lstrip('KEY:') + ' DRUG ' + field[i+3]
        else:
            fdrkey=field[i+1].lstrip('KEY:') #lstrip accounts for 'E'in fdrkey
```

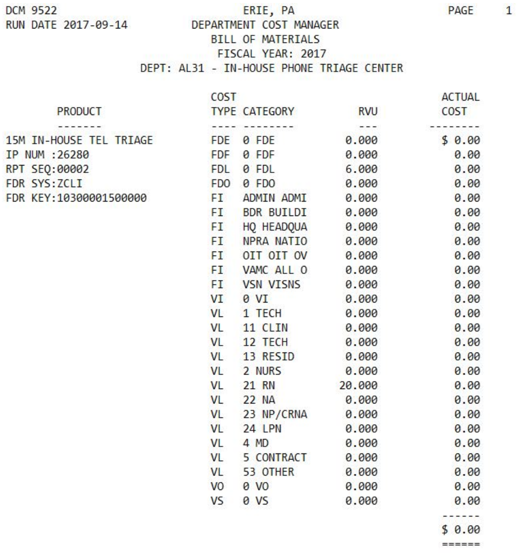
Product description

The **product description** was more challenging. There were no key words at the beginning of the line. I would have to keep track of a prior line to know when the next line would be the product description. This is where familiarity with the data was key. At the very END of each Bill of Materials was an "=" line. However, this doesn't help for the very first item.
I used 'counters' to keep track of when I came to the end of one item (the '=' line). And also to process the very first item.

Product description Code-first item

```
                if field[i] == first_var:
                    first_count = 1
                    prod=''
                    break
                #find first prod desc
                if first_count == 1:
                    #after first item update first_var
                    first_var='datasci'
                    for f in range(len(field)):
                        if field[f]=='FDE':
                      #don't want any info after FDE
                            first_count=10
                            break
                        else:
                            prod=prod+field[f]+' '
                            #redundant
                            count=0
                            first_count=10
```

9522 page 1



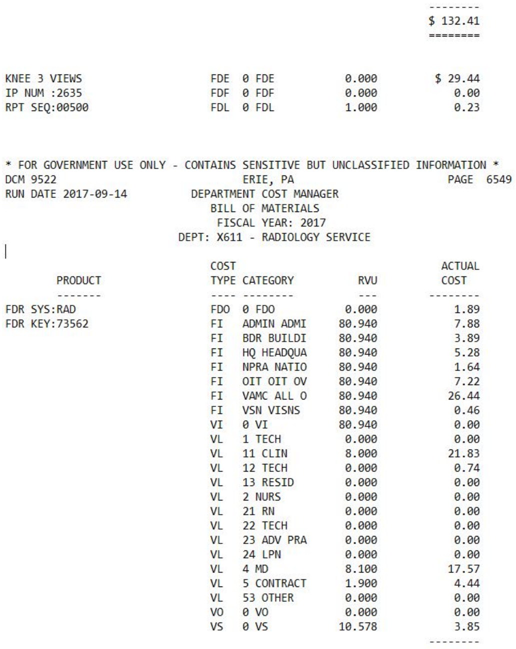
```
DCM 9522                          ERIE, PA                     PAGE     1
RUN DATE 2017-09-14         DEPARTMENT COST MANAGER
                               BILL OF MATERIALS
                                FISCAL YEAR: 2017
                     DEPT: AL31 - IN-HOUSE PHONE TRIAGE CENTER

                                COST                             ACTUAL
        PRODUCT                 TYPE CATEGORY          RVU       COST
        -------                 ---- --------          ---      --------
15M IN-HOUSE TEL TRIAGE         FDE  0 FDE            0.000     $ 0.00
IP NUM :26280                   FDF  0 FDF            0.000       0.00
RPT SEQ:00002                   FDL  0 FDL            6.000       0.00
FDR SYS:ZCLI                    FDO  0 FDO            0.000       0.00
FDR KEY:10300001500000          FI   ADMIN ADMI       0.000       0.00
                                FI   BDR BUILDI       0.000       0.00
                                FI   HQ HEADQUA       0.000       0.00
                                FI   NPRA NATIO       0.000       0.00
                                FI   OIT OIT OV       0.000       0.00
                                FI   VAMC ALL O       0.000       0.00
                                FI   VSN VISNS        0.000       0.00
                                VI   0 VI             0.000       0.00
                                VL   1 TECH           0.000       0.00
                                VL   11 CLIN          0.000       0.00
                                VL   12 TECH          0.000       0.00
                                VL   13 RESID         0.000       0.00
                                VL   2 NURS           0.000       0.00
                                VL   21 RN           20.000       0.00
                                VL   22 NA            0.000       0.00
                                VL   23 NP/CRNA       0.000       0.00
                                VL   24 LPN           0.000       0.00
                                VL   4 MD             0.000       0.00
                                VL   5 CONTRACT       0.000       0.00
                                VL   53 OTHER         0.000       0.00
                                VO   0 VO             0.000       0.00
                                VS   0 VS             0.000       0.00
                                                                ------
                                                                $ 0.00
                                                                ======
```

Figure: 9522 page 1

Product description Code-other items

```
            #find prod desc
        #'hardcode' in the startswith field[0] instead of
 using i
        if (count >0 and not field[0].startswith('*') and
 not field[0].startswith('DCM')
                and not field[0].startswith('RUN')and not
 field[0].startswith('BILL') and not
                field[0].startswith('FISCAL') and not
field[0].startswith('DEPT:') and not
                field[0].startswith('PRODUCT')and not
field[0].startswith('--------')):
            ##print ('count at not *=',count)
            ##print('field[i]=',field[i])
            ##print('field=',field)
            for d in range(len(field)):
                if field[d]=='FDE':
                    #don't want any info after FDE
                    count=0
                    break
                else:
                    prod=prod+field[d].replace(",","  ")+'
 '
```

9522 header bisects data



```
                                                                 --------
                                                                 $ 132.41
                                                                 ========


KNEE 3 VIEWS                    FDE  0 FDE          0.000        $ 29.44
IP NUM :2635                    FDF  0 FDF          0.000           0.00
RPT SEQ:00500                   FDL  0 FDL          1.000           0.23



* FOR GOVERNMENT USE ONLY - CONTAINS SENSITIVE BUT UNCLASSIFIED INFORMATION *
DCM 9522                             ERIE, PA                              PAGE  6549
RUN DATE 2017-09-14         DEPARTMENT COST MANAGER
                               BILL OF MATERIALS
                                FISCAL YEAR: 2017
                          DEPT: X611 - RADIOLOGY SERVICE
|
                                COST                                  ACTUAL
        PRODUCT                 TYPE CATEGORY           RVU           COST
        -------                 ---- --------           ---         --------
FDR SYS:RAD                     FDO  0 FDO          0.000           1.89
FDR KEY:73562                   FI   ADMIN ADMI    80.940           7.88
                                FI   BDR BUILDI    80.940           3.89
                                FI   HQ HEADQUA    80.940           5.28
                                FI   NPRA NATIO    80.940           1.64
                                FI   OIT OIT OV    80.940           7.22
                                FI   VAMC ALL O    80.940          26.44
                                FI   VSN VISNS     80.940           0.46
                                VI   0 VI          80.940           0.00
                                VL   1 TECH         0.000           0.00
                                VL   11 CLIN        8.000          21.83
                                VL   12 TECH        0.000           0.74
                                VL   13 RESID       0.000           0.00
                                VL   2 NURS         0.000           0.00
                                VL   21 RN          0.000           0.00
                                VL   22 TECH        0.000           0.00
                                VL   23 ADV PRA     0.000           0.00
                                VL   24 LPN         0.000           0.00
                                VL   4 MD           8.100          17.57
                                VL   5 CONTRACT     1.900           4.44
                                VL   53 OTHER       0.000           0.00
                                VO   0 VO           0.000           0.00
                                VS   0 VS          10.578           3.85
                                                                 --------
```

Debugging

- **Starting small** with 6,000 pages and 14,500 products (as it turns out) using the entire report to develop the code would have taken a substantial amount of time. So in the same way that I added code one item at a time, I started with a small truncated report and then used larger and larger portions of the report until I finally was using the entire report.

Debugging

- **Starting small** with 6,000 pages and 14,500 products (as it turns out) using the entire report to develop the code would have taken a substantial amount of time. So in the same way that I added code one item at a time, I started with a small truncated report and then used larger and larger portions of the report until I finally was using the entire report.
- My debugging during the main writing phase was based on the minimal **37 page** report. Once I had my code complete (or so I thought) I expanded my source data to 127 pages (looking for 'natural' product information ends at the end of a page). Again more debugging.

Debugging

- **Starting small** with 6,000 pages and 14,500 products (as it turns out) using the entire report to develop the code would have taken a substantial amount of time. So in the same way that I added code one item at a time, I started with a small truncated report and then used larger and larger portions of the report until I finally was using the entire report.
- My debugging during the main writing phase was based on the minimal **37 page** report. Once I had my code complete (or so I thought) I expanded my source data to 127 pages (looking for 'natural' product information ends at the end of a page). Again more debugging.
- The next expansion was 500 pages followed by ('surprise') more debugging. The next tests were at 1,000 then 2,000 and 3,000 pages followed by (you guessed it) more debugging.

Additional Debugging

- After the initial code was written, debugging mainly consisted of finding 'special' cases that did not conform to the 'normal' data. Most of these were relatively easily fixed by adding code that looked for these special cases. During testing I reviewed the csv file data to insure the formatting and data was accurate. This involved not only 'eyeballing' the data but also section comparison between the 9522 and csv file to insure accuracy.

Additional Debugging

- After the initial code was written, debugging mainly consisted of finding 'special' cases that did not conform to the 'normal' data. Most of these were relatively easily fixed by adding code that looked for these special cases. During testing I reviewed the csv file data to insure the formatting and data was accurate. This involved not only 'eyeballing' the data but also section comparison between the 9522 and csv file to insure accuracy.
- I then jumped to the full 6,600 page report data I was surprised to find I still had debugging to perform.

Additional Debugging

- After the initial code was written, debugging mainly consisted of finding 'special' cases that did not conform to the 'normal' data. Most of these were relatively easily fixed by adding code that looked for these special cases. During testing I reviewed the csv file data to insure the formatting and data was accurate. This involved not only 'eyeballing' the data but also section comparison between the 9522 and csv file to insure accuracy.
- I then jumped to the full 6,600 page report data I was surprised to find I still had debugging to perform.
- **Final testing** To verify that I was capturing all of the items and multiple feeder system/key pairs. I choose four 50-page blocks of the report (first 50, last 50 and two 50's in the middle). I verified textbfline by line that everything was in the csv file as it should be. This step also identified additional 'special' conditions. After accounting for these items, I again did a line by line verification for the first last 50 pages, the 50-page segment that 'failed' and then a new 50-page section.

Mis-steps and wrong directions

- Really focus on **small**: When I first started coding, even though I had in my mind to only do a little, I still did too much. I cut back and had to focus on the one item (and then adding one item at a time).

Mis-steps and wrong directions

- Really focus on **small**: When I first started coding, even though I had in my mind to only do a little, I still did too much. I cut back and had to focus on the one item (and then adding one item at a time).
- I started off using functions to read and write to the file and to extract the items, however, when I got to the product description I had to totally change: I needed to be able to save and 'remember' a variable which I can't do between functions. So I needed to switch to one continuous function.

Mis-steps and wrong directions

- Really focus on **small**: When I first started coding, even though I had in my mind to only do a little, I still did too much. I cut back and had to focus on the one item (and then adding one item at a time).
- I started off using functions to read and write to the file and to extract the items, however, when I got to the product description I had to totally change: I needed to be able to save and 'remember' a variable which I can't do between functions. So I needed to switch to one continuous function.
- **My own genius**: I am proud of myself for the way I handle finding the first line after the header (without getting any other line after the header). By defining the '—' (dash line) through a variable, I could then change the variable so it would never find the dash line again.

Final notes

- Verification debugging was the **worst**. But I knew I needed to power through it one line of data and one line of source code at a time.

Final notes

- Verification debugging was the **worst**. But I knew I needed to power through it one line of data and one line of source code at a time.
- I should have kept a notebook to keep track of everything I did. Specifically I wish I knew which pages I used for line by line checks.

Final notes

- Verification debugging was the **worst**. But I knew I needed to power through it one line of data and one line of source code at a time.
- I should have kept a notebook to keep track of everything I did. Specifically I wish I knew which pages I used for line by line checks.
- My next step will be to start using this for my rejects and get the 9522 from other stations (other VAs) to see how it works with other reports (data nomenclature is not standardized). I may need to re-write my code to focus on the FI, VL, FDL, etc. lines depending on how other hospitals enter their data.