

PROJECT

MSA Transparency Tool

Version 1.0

December 13, 2014

Medicare MSA Transparency Tool

Version History

Date	Version #	Description	Author
12/13/2014	1.0a	Final Version	Vincent Taylor

TABLE OF CONTENTS

1	INTRODUCTION	4
1.1	OBJECTIVES	4
1.2	SYSTEM DESCRIPTION	4
1.3	SCOPE	4
1.4	ISSUES.....	4
1.5	DEPENDENCIES	4
2	EXECUTION	4
2.1	SAS SERVER FILES.....	4
3	ENVIRONMENT	4
3.1	HARDWARE	4
3.2	SERVERS	4
3.3	SPECIAL TEST NEEDS	4
4	RELEASE CRITERIA.....	4
5	Source Code	6
6	Process.....	12
7	APPENDIX.....	12

1 Introduction

This work request was submitted by Marcia

1.1 Objectives

- Using CPT Codes, Revenue Codes, descriptions and MSA Fees build a master file.
- Initially create four query program.
- Deliver too to end user.
- Train Medicare Customer Service Hccs on use of the tool.

1.2 System Description

System/Application Name	Decision Support
Director Decision Support	Bev
VP & GM Medicare Division	Marcia
Manager Medicare Customer Srv	Jeannie

1.3 Scope

Design a reporting tool for Medicare which will allow Medicare Customer Service HCCs to provide fee schedule information to NPP members calling with various requests.

1.4 Issues

None foreseen at this time.

1.5 Dependencies

N/A.

2 Execution

2.1 SAS Web Report Studio, Stored Process Web Application and Information Delivery Portal

3 Environment

3.1 Hardware

Laptop and Desktop work-stations

3.2 Servers

SAS Enterprise affsanfe01
SAS Enterprise affsass00p01

3.3 Special Test Needs

Refer to Testing Case Selection

4 Release Criteria

- System testing, independent testing and user acceptance testing must be carried out successfully
- Testing of all requirement enhancements, if any in the current release, must be completed

Medicare MSA Transparency Tool

- SAS Source Code

Member Provider History Report	Completed
MSA Transparency Tool	Completed
Speciality City Report	Completed
Speciality Revenue Description Report	Completed
Speciality County Report	Completed

Note: MSA Transparency Tool program is used to build the master input file it will require little to no modifications in the future.

MSA Transparency Tool

```
/*-----*/
/* Timer Start */
/*-----*/
data _null_;
    time = time();
    call symput('Start',time);
run;

options validvarname=v7 source2 symbolgen mautosource mprint sasautos=(m)
mlogic mrecall;

/*-----*/
/* Library \ file Definitions */
/*-----*/

%let DRIVE_I=\\affsanfe01\Aff_Departments\Departments_Network_Health; * I
Drive;
%let DRIVE_Y=\\affsanfe01\Share_Team; * Y Drive;
%let DRIVE_G=\\affsassoc00p01;*SAS server;

/* Global Variable Definitions */

%global DRIVE_I DRIVE_Y DRIVE_G;

%let fdate = %sysfunc(today(),date7.);

%let YR = %substr(&fdate,6,2);
%let begdate = "01JAN20&YR"d;
%let enddate = "31DEC20&YR"d;

%put &begdate &enddate;

libname star "&DRIVE_G\sasdata\SAS Warehouse\Star";

libname goldmine "&DRIVE_Y\Disease_Mgmt_Online\Medicare Chronic Care
Improvement Program\Medicare Chronic Care Improvement Program\
```

Medicare MSA Transparency Tool

```
Base populations and new members\DS\Goldmine";

libname WrkBk Excel "&DRIVE_I.\DecisionSupport\vtaylor\MSA Transparency
Tool\MSACPTCodes.xlsx";
libname MSA "&DRIVE_I.\DecisionSupport\vtaylor\MSA Transparency Tool";

proc datasets library=work memtype=data kill nolist;
run;

/* Get Billable Revenue Codes and Descriptions */

data file1;
    format cpt_code $5.;
    set WrkBk.'REVCODES$'n;
    cpt_code = substr(cpt_codes_description,1,5);
run;

DATA MSA.Revcodes (drop=f6);
    set file1;
    if cpt_code = ' ' then delete;
run;
proc sort data=MSA.revcodes;
    by cpt_code;

    /* Get Billable Revenue Codes and Medicare Fees */

data file2;
    format cpt_code $5.;
    set WrkBk.'FEES$'n;
    cpt_code = substr(cpt_codes_description,1,5);
run;
DATA MSA.fees;
    set file2;
run;

proc sort data=MSA.fees;
    by cpt_code;

/* Merge Billable Revenue Codes, Descriptions and Fees */
proc sql;
    create table MSA.revcode_desc_fees_table as
        select a.*,
               b.Medicare_Fee_Urban_Rate,
               b.Medicare_Fee_Rural_Rate

        from MSA.revcodes      a,
             MSA.fees          b

        where a.cpt_code = b.cpt_code;
quit;

proc sort data=MSA.revcode_desc_fees_table;
    by cpt_code rev_code;
```

Medicare MSA Transparency Tool

```

libname WrkBk clear;

/* Create NPP Membership Table */

proc sql;
  create table npp_membership as
    select distinct
      a.member_id              as SUBSCRIBER_ID,
      a.product_key,
      a.pcp_key,
      b.benefit_plan_code      as PLAN_ID,
      b.benefit_plan_desc      as PLAN_DESC,
      c.first_name             as MBR_FIRST_NAME,
      c.last_name              as MBR_LAST_NAME,
      c.state                  as MBR_STATE,
      c.address1               as HOME_ADDR1,
      c.address2               as HOME_ADDR2,
      c.city                   as HOME_CITY,
      c.state                  as HOME_STATE,
      c.zip_code               as HOME_ZIP,
      c.county                 as HOME_COUNTY,
      c.phone                  as HOME_PHONE,
      c.social_security_number as TAXID
    from   star.member_month_fact a,
           star.product_dim       b,
           star.member_dim        c

    where  a.product_key = b.product_key and
           a.member_id   = c.member_id   and
           &begdate<=a.beg_date<=&enddate and
           b.lob_code in ('NPP');

quit;

proc sort data=npp_membership nodupkey;
  by subscriber_id;

/* Get Goldmine deceased members. This is a validation step, member count
should remain unchanged */
data term_members (rename =(member_id = SUBSCRIBER_ID));
  set goldmine.goldmine_sas;
run;

proc sort data=term_members;
  by subscriber_id;

data MSA.npp_members (drop=hic_number last_name first_name input_term_date
term_reason phone
                        input_effective_date mail_add_1 mail_add_2
mail_add_3 mail_city
                        mail_state mail_zip app_date batch term_date
effective_date rundate
                        qnxt_member_id address1 address2 address3 city state
zip);

```

Medicare MSA Transparency Tool

```

merge npp_membership (in=a) term_members (in=b);
  by subscriber_id;
  if ^b;
run;

/* Read Claim Detail Fact and Claim Provider Dim Warehouse SAS Dataset */

proc sql;
  create table claim_provider_facts as
    select a.procedure_key,
           a.pcp_key,
           a.document_number           as CLAIM_ID,
           a.line_item_number          as CLAIM_LINE_NBR,
           a.member_id                 as SUBSCRIBER_ID,
           a.beg_date                  as RECVD_DT,
           a.quantity                   as UNIT,
           a.billed_amt                as CHARGED_AMOUNT,
           a.allowed_amt               as ALLOWED_AMOUNT,
           a.coinsurance_amt           as COINS_AMT,
           a.COPAY_AMT,
           a.PAID_AMT,
           a.payment_date              as PAID_CLAIM_DATE,
           b.claim_prov_id             as PROVIDER_ID,
           b.claim_provider            as PROVIDER_NAME,
           b.claim_prov_phy_addr1      as SRVC_ADDR1,
           b.claim_prov_phy_addr2      as SRVC_ADDR2,
           b.claim_prov_phy_city       as SRVC_CITY,
           b.claim_prov_phy_state      as SRVC_STATE,
           b.claim_prov_phy_zip        as SRVC_ZIP,
           b.claim_prov_county         as SRVC_COUNTY,
           b.claim_prov_phone          as SRVC_PHONE,
           b.claim_npi                 as NPI,
           b.claim_prov_spec_desc      as SPECIALTY,
           c.first_dx_code             as DIAGNOSIS,
           c.first_dx_desc             as DIAGNOSIS_DESC,
           d.bill_type_code1           as FACILITY_BILL_TYPE_CODE1,
           d.bill_type_desc1           as
FACILITY_BILL_TYPE_DESC1,
           d.bill_type_code2           as FACILITY_BILL_TYPE_CODE2,
           d.bill_type_desc2           as FACILITY_BILL_TYPE_DESC2,
           e.adjustment_code           as CLM_DENIAL_CODE,
           e.adjustment_desc           as CLM_DENIAL_DESC,
           f.PLACE_OF_SERVICE_DESC     as PLACE_OF_SRVC_DESC

    from   star.claim_detail_fact      a,
           star.claim_provider_dim    b,
           star.first_dx_dim          c,
           star.hospital_dim          d,
           star.adjudication_dim      e,
           star.setting_dim           f

    where  a.claim_prov_key           = b.claim_prov_key   and
           a.first_dx_key             = c.first_dx_key     and

```


Medicare MSA Transparency Tool

```

a.adjudication_key = e.adjudication_key and
a.hospital_key     = d.hospital_key     and
a.setting_key      = f.setting_key      and
f.facility_ind     = 'Y'                and
a.lob_code in ('NPP')                   and
&begdate<=a.beg_date<=&enddate ;

quit;

proc sort data=claim_provider_facts;
  by claim_id provider_name;

/* Qualify previous selected providers by cpt codes which match */

proc sql;
  create table match_claim_providers_facts as
    select a.*,
           b.PROCEDURE_CODE,
           b.PROCEDURE_DESC,
           b.procedure_modifier_code as MODIFIER_CODE_1,
           b.procedure_modifier2_code as MODIFIER_CODE_2,
           c.*
    from   claim_provider_facts      a,
           star.procedure_dim        b,
           MSA.revcode_desc_fees_table c
    where  a.procedure_key = b.procedure_key and
           b.cpt_code      = c.cpt_code      and
           b.rev_code      = c.rev_code      ;

quit;

proc sort data=match_claim_providers_facts out=match_cpfs;
  by cpt_code rev_code;

/* Qualify previous selected providers by cpt codes which do not match */

data nonmatch_claim_providers_facts;
  merge match_cpfs (in=a) MSA.revcode_desc_fees_table (in=b);
  by cpt_code rev_code;

  if (b and a) then;
  else output;

run;

proc sort data=nonmatch_claim_providers_facts nodupkey;
  by cpt_code rev_code;

data MSA.qualified_claim_facility (drop=hpcd procedure_key claim_line_nbr unit
paid_claim_date np1);
  format hpcd $60.;
  set match_cpfs
      nonmatch_claim_providers_facts;
  if procedure_key =. or procedure_key = "" or procedure_key = ' ' then
delete;
  hpcd = translate(procedure_desc, ' ', '-');

```

Medicare MSA Transparency Tool

```
procedure_desc = hpcd;  
hpcd = translate(procedure_desc, ' ', '/');  
procedure_desc = hpcd;  
run;
```

```
proc sort data=MSA.qualified_claim_facility;  
  by claim_id subscriber_id provider_name cpt_code rev_code;
```

Note: Member Provider History program uses as its parm input a specific Subscriber Id. This program can be used to provide a subscriber with claim information from the current calendar year for all seen providers. This program can be modified should additional claims information be required.

Member Provider History Report

```
*ProcessBody;
```

```
%global SUBSCRIBER_PARM;  
%STPBEGIN;
```

```
/*-----*/  
/* Timer Start */  
/*-----*/
```

```
data _null_;  
  time = time();  
  call symput('Start',time);  
run;
```

```
options validvarname=v7;
```

```
/*-----*/  
/* Library \ file Definitions */  
/*-----*/
```

```
%let DRIVE_I=\\affsanfe01\Aff_Departments\Departments_Network_Health; * I  
Drive;
```

```
%let DRIVE_Y=\\affsanfe01\Share_Team; * Y Drive;
```

```
%let DRIVE_G=\\affsassoc01\SAS server;
```

```
/* Global Variable Definitions */
```

```
%global DRIVE_I DRIVE_Y DRIVE_G SUBSCRIBER;
```

```
libname star "&DRIVE_G\sasdata\SAS Warehouse\Star";
```

```
libname MSA "&DRIVE_I.\DecisionSupport\vtaylor\MSA Transparency Tool";
```

```
proc datasets library=work memtype=data kill nolist;  
run;
```

```
/* Get Qualified Claim Provider SAS Dataset */
```

```
DATA memprov (keep=CLAIM_ID SUBSCRIBER_ID CHARGED_AMOUNT ALLOWED_AMOUNT  
COINS_AMT COPAY_AMT PAID_AMT PROVIDER_NAME SRVC_ADDR1 SRVC_CITY SRVC_STATE
```

Medicare MSA Transparency Tool

```
SRVC_ZIP SRVC_PHONE SPECIALTY CPT_CODES_DESCRIPTION REV_CODE
REVENUE_CODES_DESCRIPTION MEDICARE_FEE_URBAN_RATE MEDICARE_FEE_RURAL_RATE);

    set MSA.qualified_claim_facility ;

    if &SUBSCRIBER_PARM = SUBSCRIBER_ID then output;

run;

proc sort data=memprow;
    by provider_name;

options nocenter ps=100 ls=160;
proc report data=memprow headline split='*';

title1 '                Network Health';
title2 'Summary of Member Provider History Report';

column CLAIM_ID SUBSCRIBER_ID PROVIDER_NAME SRVC_ADDR1 SRVC_CITY SRVC_STATE
SRVC_ZIP SRVC_PHONE
        SPECIALTY CPT_CODES_DESCRIPTION REV_CODE REVENUE_CODES_DESCRIPTION
MEDICARE_FEE_URBAN_RATE
        MEDICARE_FEE_RURAL_RATE CHARGED_AMOUNT ALLOWED_AMOUNT COINS_AMT
COPAY_AMT PAID_AMT;

define claim_id / group 'Claim*Id' width=7;
define subscriber_id / group 'Subscriber*Id' width=12;

define provider_name / group 'Provider*Name' width=12;
define srvc_addr1 / group 'Service*Address' width=15;
define srvc_city / group 'Service*City' width=11;
define srvc_state / group 'Service*State' width=12;
define srvc_zip / group 'Service*Zip' width=10;
define srvc_phone / group 'Service*phone' width=12;
define specialty / group 'Specialty' width=9;
define cpt_codes_description / group 'CPT_CODE*Description' width=19;
define rev_code / group 'REV*CODE' width=7;
define revenue_codes_description / group 'REV_CODE*Description' width=19;
define medicare_fee_urban_rate / group 'Medicare_Fee*Urban_Rate' width=22;
define medicare_fee_rural_rate / display 'Medicare_Fee*Rural_Rate' width=22;
define charged_amount / group 'Charge*Amount' width=12;
define allowed_amount / group 'Allowed*Amount' width=14;
define coins_amt / group 'Coins*Amount' width=12;
define copay_amt / group 'Copay*Amount' width=12;
define paid_amt / group 'Paid*Amount' width=10;

run;

%STPEND;
```

Medicare MSA Transparency Tool

Note: Speciality City program uses as its parm input the 1st name for a specific specialist, ie. 'internal' for 'Internal Medicine'; and the 1st four letters of a specific city, ie. 'App' for 'Appleton' in which the specialist resides. The program can be used to provide a subscriber with the name, address, phone, cpt code, cpt code description, revenue code, revenue code description, MSA fee, coinsurance, copayment, and allowed amounts for providers. The program can be modified should additional information, i.e such as a report per 'county' rather than 'city' be required.

```
%global SPECIALTY_NAME SERVICE_PROVIDER_CITY SERVICE_PROVIDER_NAME
SERVICE_PLACE_DESCRIPTION
        SERVICE_PROCEDURE_DESCRIPTION ;

%STPBEGIN;

/*-----*/
/* Timer Start */
/*-----*/
data _null_;
    time = time();
    call symput('Start',time);
run;

options validvarname=v7;

/*-----*/
/* Library \ file Definitions */
/*-----*/

%let DRIVE_I=\\affsanfe01\Aff_Departments\Departments_Network_Health; * I
Drive;
%let DRIVE_Y=\\affsanfe01\Share_Team; * Y Drive;
%let DRIVE_G=\\affsass00p01;*SAS server;

/* Global Variable Definitions */

%global DRIVE_I DRIVE_Y DRIVE_G T1;

libname star "&DRIVE_G\sasdata\SAS Warehouse\Star";

libname MSA "&DRIVE_I.\DecisionSupport\vtaylor\MSA Transparency Tool";

proc datasets library=work memtype=data kill nolist;
run;

%macro process;
/* Select Specialty that Sound like */
%if (&SPECIALTY_NAME NE %STR( ) and &SERVICE_PROVIDER_CITY NE %STR( )) and
    (&SERVICE_PROVIDER_NAME EQ %STR( ) and &SERVICE_PLACE_DESCRIPTION EQ %STR(
) and
    &SERVICE_PROCEDURE_DESCRIPTION EQ %STR( ))
%then
%do;
    %let t1 = %str(Summary of Specialty by City Report);
```

Medicare MSA Transparency Tool

```

proc sql;
    create table specialty_tbl as
    select a.ALLOWED_AMOUNT,
           a.COINS_AMT,
           a.COPAY_AMT,
           a.PROVIDER_NAME,
           a.SRVC_ADDR1,
           a.SRVC_CITY,
           a.SRVC_STATE,
           a.SRVC_ZIP,
           a.SRVC_PHONE,
           a.SRVC_COUNTY,
           a.SPECIALTY,
           a.PLACE_OF_SRVC_DESC,
           a.CPT_CODES_DESCRIPTION,
           a.REV_CODE,
           a.REVENUE_CODES_DESCRIPTION,
           a.MEDICARE_FEE_URBAN_RATE,
           a.MEDICARE_FEE_RURAL_RATE

    from   MSA.qualified_claim_facility          a

    where  index(UPCASE(a.SPECIALTY),UPCASE("&SPECIALTY_NAME")) > 0 and

index(UPCASE(a.SRVC_CITY),UPCASE("&SERVICE_PROVIDER_CITY")) > 0
    ;
quit;
%end;

%if (&SPECIALTY_NAME NE %STR( ) and &SERVICE_PROVIDER_CITY NE %STR( )) and
    (&SERVICE_PROVIDER_NAME NE %STR( ) and &SERVICE_PLACE_DESCRIPTION EQ %STR(
) and
    &SERVICE_PROCEDURE_DESCRIPTION EQ %STR( ))
%then
%do;
    %let t1 = %str(Summary of Specialty by Provider Name and City Report);
proc sql;
    create table specialty_tbl as
    select a.ALLOWED_AMOUNT,
           a.COINS_AMT,
           a.COPAY_AMT,
           a.PROVIDER_NAME,
           a.SRVC_ADDR1,
           a.SRVC_CITY,
           a.SRVC_STATE,
           a.SRVC_ZIP,
           a.SRVC_PHONE,
           a.SRVC_COUNTY,
           a.SPECIALTY,
           a.PLACE_OF_SRVC_DESC,
           a.CPT_CODES_DESCRIPTION,
           a.REV_CODE,
           a.REVENUE_CODES_DESCRIPTION,
           a.MEDICARE_FEE_URBAN_RATE,

```

Medicare MSA Transparency Tool

```

a.MEDICARE_FEE_RURAL_RATE

from MSA.qualified_claim_facility a

where index(UPCASE(a.SPECIALTY),UPCASE("&SPECIALTY_NAME")) > 0
and
index(UPCASE(a.SRVC_CITY),UPCASE("&SERVICE_PROVIDER_CITY")) > 0 and
index(UPCASE(a.PROVIDER_NAME),UPCASE("&SERVICE_PROVIDER_NAME")) > 0
;
quit;
%end;

%if (&SPECIALTY_NAME NE %STR( ) and &SERVICE_PROVIDER_CITY NE %STR( )) and
(&SERVICE_PROVIDER_NAME NE %STR( ) and &SERVICE_PLACE_DESCRIPTION NE %STR(
) and
&SERVICE_PROCEDURE_DESCRIPTION EQ %STR( ))
%then
%do;
    %let t1 = %str(Summary of Specialty by Provider Name, Place of Service and
City Report);
    proc sql;
        create table specialty_tbl as
        select a.ALLOWED_AMOUNT,
               a.COINS_AMT,
               a.COPAY_AMT,
               a.PROVIDER_NAME,
               a.SRVC_ADDR1,
               a.SRVC_CITY,
               a.SRVC_STATE,
               a.SRVC_ZIP,
               a.SRVC_PHONE,
               a.SRVC_COUNTY,
               a.SPECIALTY,
               a.PLACE_OF_SRVC_DESC,
               a.CPT_CODES_DESCRIPTION,
               a.REV_CODE,
               a.REVENUE_CODES_DESCRIPTION,
               a.MEDICARE_FEE_URBAN_RATE,
               a.MEDICARE_FEE_RURAL_RATE

               from MSA.qualified_claim_facility a

               where index(UPCASE(a.SPECIALTY),UPCASE("&SPECIALTY_NAME")) > 0
and
index(UPCASE(a.SRVC_CITY),UPCASE("&SERVICE_PROVIDER_CITY")) > 0 and
index(UPCASE(a.PROVIDER_NAME),UPCASE("&SERVICE_PROVIDER_NAME")) > 0 and
index(UPCASE(a.PLACE_OF_SRVC_DESC),UPCASE("&SERVICE_PLACE_DESCRIPTION")) > 0
;
quit;

```

Medicare MSA Transparency Tool

```

%end;

%if (&SPECIALTY_NAME NE %STR( ) and &SERVICE_PROVIDER_CITY NE %STR( )) and
    (&SERVICE_PROVIDER_NAME NE %STR( ) and &SERVICE_PLACE_DESCRIPTION NE %STR(
) and
    &SERVICE_PROCEDURE_DESCRIPTION NE %STR( ))
%then
%do;
    %let t1 = %str(Summary of Specialty by Provider Name, Place of Service,
Procedure Description and City Report);
    proc sql;
        create table specialty_tbl as
        select a.ALLOWED_AMOUNT,
            a.COINS_AMT,
            a.COPAY_AMT,
            a.PROVIDER_NAME,
            a.SRVC_ADDR1,
            a.SRVC_CITY,
            a.SRVC_STATE,
            a.SRVC_ZIP,
            a.SRVC_PHONE,
                a.SRVC_COUNTY,
            a.SPECIALTY,
                a.PLACE_OF_SRVC_DESC,
            a.CPT_CODES_DESCRIPTION,
            a.REV_CODE,
            a.REVENUE_CODES_DESCRIPTION,
            a.MEDICARE_FEE_URBAN_RATE,
            a.MEDICARE_FEE_RURAL_RATE

            from MSA.qualified_claim_facility a

            where index(UPCASE(a.SPECIALTY),UPCASE("&SPECIALTY_NAME")) > 0
and
index(UPCASE(a.SRVC_CITY),UPCASE("&SERVICE_PROVIDER_CITY")) > 0
and
index(UPCASE(a.PROVIDER_NAME),UPCASE("&SERVICE_PROVIDER_NAME")) > 0
and
index(UPCASE(a.PLACE_OF_SRVC_DESC),UPCASE("&SERVICE_PLACE_DESCRIPTION")) > 0
and
index(UPCASE(a.PROCEDURE_DESC),UPCASE("&SERVICE_PROCEDURE_DESCRIPTION")) > 0
        ;
    quit;
%end;

%mend process;

%process;

proc sort data=specialty_tbl out=specialty_out noduprecs;

```

Medicare MSA Transparency Tool

```
by _all_;
run;

data specialty_rpt;
  set specialty_out;
run;

options nocenter ps=100;
proc report data=specialty_rpt headline split='*' missing spacing=0;

title1 'Network Health';
title2 &T1;

column SRVC_ZIP PROVIDER_NAME SRVC_CITY SRVC_ADDR1 SRVC_COUNTY SRVC_STATE
SRVC_PHONE SPECIALTY
        PLACE_OF_SRVC_DESC CPT_CODES_DESCRIPTION REV_CODE
REVENUE_CODES_DESCRIPTION MEDICARE_FEE_URBAN_RATE
        MEDICARE_FEE_RURAL_RATE ALLOWED_AMOUNT COINS_AMT COPAY_AMT ;

define srvc_zip / order order=data 'Service*Zip' width=10;
define provider_name / group 'Provider*Name' width=12;
define srvc_city / group 'Service*City' width=11;
define srvc_addr1 / group 'Service*Address' width=15;
define srvc_county / group 'Service*County' width=13;
define srvc_state / group 'Service*State' width=12;
define srvc_phone / group 'Service*phone' width=12;
define specialty / group 'Specialty' width=9;
define place_of_srvc_desc / group 'Place_of*Service_Description' width=28;
define cpt_codes_description / display 'CPT_CODE*Description' width=19;
define rev_code / display 'REV*CODE' width=7;
define revenue_codes_description / display 'REV_CODE*Description' width=19;
define medicare_fee_urban_rate / display 'Medicare_Fee*Urban_Rate' width=22;
define medicare_fee_rural_rate / display 'Medicare_Fee*Rural_Rate' width=22;
define allowed_amount / display 'Allowed*Amount' width=14;
define coins_amt / display 'Coins*Amount' width=12;
define copay_amt / display 'Copay*Amount' width=12;

run;

%STPEND;
```


Medicare MSA Transparency Tool

Note: Speciality County program uses as its parm input the 1st name for a specific specialist, ie. 'internal' for 'Internal Medicine'; and the 1st four letters of a specific count, ie. 'Brow' for 'Brown' in which the specialist resides. The program can be used to provide a subscriber with the name, address, phone, cpt code, cpt code description, revenue code, revenue code description, MSA fee, coinsurance, copayment, and allowed amounts for providers. The program can be modified should additional information, i.e. such as a report per 'county' rather than 'city' be required.

Specialty County Report

```
*Processbody;

%global SPECIALTY_NAME SERVICE_PROVIDER_COUNTY SERVICE_PROVIDER_NAME
SERVICE_PLACE_DESCRIPTION
        SERVICE_PROCEDURE_DESCRIPTION ;

%STPBEGIN;

/*-----*/
/* Library \ file Definitions */
/*-----*/

%let DRIVE_I=\\affsanfe01\Aff_Departments\Departments_Network_Health; * I
Drive;
%let DRIVE_Y=\\affsanfe01\Share_Team; * Y Drive;
%let DRIVE_G=\\affsassy00p01;*SAS server;

/* Global Variable Definitions */

%global DRIVE_I DRIVE_Y DRIVE_G T1;

libname star "&DRIVE_G\sasdata\SAS Warehouse\Star";

libname MSA "&DRIVE_I.\DecisionSupport\vtaylor\MSA Transparency Tool";

proc datasets library=work memtype=data kill nolist;
run;

%macro process;
/* Select Specialty that Sound like */
%if (&SPECIALTY_NAME NE %STR( ) and &SERVICE_PROVIDER_COUNTY NE %STR( )) and
    (&SERVICE_PROVIDER_NAME EQ %STR( ) and &SERVICE_PLACE_DESCRIPTION EQ %STR(
) and
    &SERVICE_PROCEDURE_DESCRIPTION EQ %STR( ))
%then
%do;
    %let t1 = %str(Summary of Specialty by County Report);
    proc sql;
        create table specialty_tbl as
        select a.ALLOWED_AMOUNT,
               a.COINS_AMT,
               a.COPAY_AMT,
               a.PROVIDER_NAME,
```

Medicare MSA Transparency Tool

```

a.SRVC_ADDR1,
a.SRVC_CITY,
a.SRVC_STATE,
a.SRVC_ZIP,
a.SRVC_PHONE,
    a.SRVC_COUNTY,
a.SPECIALTY,
    a.PLACE_OF_SRVC_DESC,
    a.REV_CODE,
a.CPT_CODES_DESCRIPTION,
a.REVENUE_CODES_DESCRIPTION,
a.MEDICARE_FEE_URBAN_RATE,
a.MEDICARE_FEE_RURAL_RATE

from    MSA.qualified_claim_facility          a

where   index(UPCASE(a.SPECIALTY),UPCASE("&SPECIALTY_NAME")) > 0 and

index(UPCASE(a.SRVC_COUNTY),UPCASE("&SERVICE_PROVIDER_COUNTY")) > 0

;
quit;
%end;

%if (&SPECIALTY_NAME NE %STR( ) and &SERVICE_PROVIDER_COUNTY NE %STR( )) and
    (&SERVICE_PROVIDER_NAME NE %STR( ) and &SERVICE_PLACE_DESCRIPTION EQ %STR(
) and
    &SERVICE_PROCEDURE_DESCRIPTION EQ %STR( ))
%then
%do;
    %let t1 = %str(Summary of Specialty by Provider Name and County
Report);
    proc sql;
        create table specialty_tbl as
        select a.ALLOWED_AMOUNT,
            a.COINS_AMT,
            a.COPAY_AMT,
            a.PROVIDER_NAME,
            a.SRVC_ADDR1,
            a.SRVC_CITY,
            a.SRVC_STATE,
            a.SRVC_ZIP,
            a.SRVC_PHONE,
                a.SRVC_COUNTY,
            a.SPECIALTY,
                a.PLACE_OF_SRVC_DESC,
            a.CPT_CODES_DESCRIPTION,
                a.REV_CODE,
            a.REVENUE_CODES_DESCRIPTION,
            a.MEDICARE_FEE_URBAN_RATE,
            a.MEDICARE_FEE_RURAL_RATE

from    MSA.qualified_claim_facility          a

```

Medicare MSA Transparency Tool

```

        where index(UPCASE(a.SPECIALTY),UPCASE("&SPECIALTY_NAME")) > 0
and
index(UPCASE(a.SRVC_COUNTY),UPCASE("&SERVICE_PROVIDER_COUNTY")) > 0 and
index(UPCASE(a.PROVIDER_NAME),UPCASE("&SERVICE_PROVIDER_NAME")) > 0
    ;
quit;
%end;

%if (&SPECIALTY_NAME NE %STR( ) and &SERVICE_PROVIDER_COUNTY NE %STR( )) and
    (&SERVICE_PROVIDER_NAME NE %STR( ) and &SERVICE_PLACE_DESCRIPTION NE %STR(
) and
    &SERVICE_PROCEDURE_DESCRIPTION EQ %STR( ))
%then
%do;
    %let t1 = %str(Summary of Specialty by Provider Name, Place of Service and
County Report);
    proc sql;
        create table specialty_tbl as
        select a.ALLOWED_AMOUNT,
            a.COINS_AMT,
            a.COPAY_AMT,
            a.PROVIDER_NAME,
            a.SRVC_ADDR1,
            a.SRVC_CITY,
            a.SRVC_STATE,
            a.SRVC_ZIP,
            a.SRVC_PHONE,
            a.SRVC_COUNTY,
            a.SPECIALTY,
            a.PLACE_OF_SRVC_DESC,
            a.CPT_CODES_DESCRIPTION,
            a.REV_CODE,
            a.REVENUE_CODES_DESCRIPTION,
            a.MEDICARE_FEE_URBAN_RATE,
            a.MEDICARE_FEE_RURAL_RATE

            from MSA.qualified_claim_facility a

            where index(UPCASE(a.SPECIALTY),UPCASE("&SPECIALTY_NAME")) > 0
and
index(UPCASE(a.SRVC_COUNTY),UPCASE("&SERVICE_PROVIDER_COUNTY")) > 0 and
index(UPCASE(a.PROVIDER_NAME),UPCASE("&SERVICE_PROVIDER_NAME")) > 0 and
index(UPCASE(a.PLACE_OF_SRVC_DESC),UPCASE("&SERVICE_PLACE_DESCRIPTION")) > 0
    ;
quit;
%end;

%if (&SPECIALTY_NAME NE %STR( ) and &SERVICE_PROVIDER_COUNTY NE %STR( )) and

```

Medicare MSA Transparency Tool

```

    (&SERVICE_PROVIDER_NAME NE %STR( ) and &SERVICE_PLACE_DESCRIPTION NE %STR(
) and
    &SERVICE_PROCEDURE_DESCRIPTION NE %STR( ))
%then
%do;
    %let t1 = %str(Summary of Specialty by Provider Name, Place of Service,
Procedure Description and County Report);
    proc sql;
        create table specialty_tbl as
        select a.ALLOWED_AMOUNT,
              a.COINS_AMT,
              a.COPAY_AMT,
              a.PROVIDER_NAME,
              a.SRVC_ADDR1,
              a.SRVC_CITY,
              a.SRVC_STATE,
              a.SRVC_ZIP,
              a.SRVC_PHONE,
              a.SRVC_COUNTY,
              a.SPECIALTY,
              a.PLACE_OF_SRVC_DESC,
              a.CPT_CODES_DESCRIPTION,
              a.REV_CODE,
              a.REVENUE_CODES_DESCRIPTION,
              a.MEDICARE_FEE_URBAN_RATE,
              a.MEDICARE_FEE_RURAL_RATE

        from   MSA.qualified_claim_facility          a

        where  index(UPCASE(a.SPECIALTY),UPCASE("&SPECIALTY_NAME")) > 0

and

        index(UPCASE(a.SRVC_COUNTY),UPCASE("&SERVICE_PROVIDER_COUNTY")) > 0
and

        index(UPCASE(a.PROVIDER_NAME),UPCASE("&SERVICE_PROVIDER_NAME")) > 0
and

        index(UPCASE(a.PLACE_OF_SRVC_DESC),UPCASE("&SERVICE_PLACE_DESCRIPTION")) > 0
and

        index(UPCASE(a.PROCEDURE_DESC),UPCASE("&SERVICE_PROCEDURE_DESCRIPTION")) > 0
        ;
    quit;
%end;

%mend process;

%process;

proc sort data=specialty_tbl out=specialty_out noduprecs;
    by _all_;
run;

```

Medicare MSA Transparency Tool

```
data specialty_rpt;
    set specialty_out;
run;

options nocenter ps=100;
proc report data=specialty_rpt headline split='*' missing spacing=0;

title1 'Network Health';
title2 &T1;

column SRVC_ZIP PROVIDER_NAME SRVC_CITY SRVC_ADDR1 SRVC_COUNTY SRVC_STATE
SRVC_PHONE SPECIALTY
        PLACE_OF_SRVC_DESC CPT_CODES_DESCRIPTION REV_CODE
REVENUE_CODES_DESCRIPTION MEDICARE_FEE_URBAN_RATE
        MEDICARE_FEE_RURAL_RATE ALLOWED_AMOUNT COINS_AMT COPAY_AMT ;

define srvc_zip / order order=data 'Service*Zip' width=10;
define provider_name / group 'Provider*Name' width=12;
define srvc_city / group 'Service*City' width=11;
define srvc_addr1 / group 'Service*Address' width=15;
define srvc_county / group 'Service*County' width=13;
define srvc_state / group 'Service*State' width=12;
define srvc_phone / group 'Service*phone' width=12;
define specialty / group 'Specialty' width=9;
define place_of_srvc_desc / group 'Place_of*Service_Description' width=28;
define cpt_codes_description / display 'CPT_CODE*Description' width=19;
define rev_code / display 'REV*CODE' width=7;
define revenue_codes_description / display 'REV_CODE*Description' width=19;
define medicare_fee_urban_rate / display 'Medicare_Fee*Urban_Rate' width=22;
define medicare_fee_rural_rate / display 'Medicare_Fee*Rural_Rate' width=22;
define allowed_amount / display 'Allowed*Amount' width=14;
define coins_amt / display 'Coins*Amount' width=12;
define copay_amt / display 'Copay*Amount' width=12;

run;

%STPEND;
```

Medicare MSA Transparency Tool

Note: Speciality CPT Code program uses as its parm input a specific five digit cpt code and the 1st four letters of a specific city, ie. 'App' for 'Appleton' in which the specialist resides. The program can be used to provide a subscriber with the name, address, phone, cpt code, cpt code description, revenue code, revenue code description, MSA fee, coinsurance, copayment, and allowed amounts for providers. The program can be modified should additional information, i.e. such as a report per 'revenue code' rather than 'cpt code' be required.

Specialty CPT Code Report

```
*ProcessBody;

%global SPECIALTY_CPTCODE SERVICE_PROVIDER_CITY SERVICE_PROVIDER_NAME
SERVICE_PLACE_DESCRIPTION
SERVICE_PROCEDURE_DESCRIPTION ;
%STPBEGIN;

/*-----*/
/* Timer Start */
/*-----*/
data _null_;
    time = time();
    call symput('Start',time);
run;

options validvarname=v7;

/*-----*/
/* Library \ file Definitions */
/*-----*/

%let DRIVE_I=\\affsanfe01\Aff_Departments\Departments_Network_Health; * I
Drive;
%let DRIVE_Y=\\affsanfe01\Share_Team; * Y Drive;
%let DRIVE_G=\\affsassoc00p01;*SAS server;

/* Global Variable Definitions */

%global DRIVE_I DRIVE_Y DRIVE_G SPECIALTY_CPTCODE SERVICE_PROVIDER_CITY
SERVICE_PROVIDER_NAME
SERVICE_PLACE_DESCRIPTION SERVICE_PROCEDURE_DESCRIPTION T1;

libname star "&DRIVE_G\sasdata\SAS Warehouse\Star";

libname MSA "&DRIVE_I.\DecisionSupport\vtaylor\MSA Transparency Tool";

proc datasets library=work memtype=data kill nolist;
run;

%macro process;
/* Select Specialty that Sound like */
%if (&SPECIALTY_CPTCODE NE %STR( ) and &SERVICE_PROVIDER_CITY NE %STR( ))
and
```

Medicare MSA Transparency Tool

```

(&SERVICE_PROVIDER_NAME EQ %STR( ) and &SERVICE_PLACE_DESCRIPTION EQ %STR(
) and
&SERVICE_PROCEDURE_DESCRIPTION EQ %STR( ))
%then
%do;
    %let t1 = %str(Summary of CPT CODE by Specific City Report);
    proc sql;
        create table specialty_tbl as
        select a.ALLOWED_AMOUNT,
              a.COINS_AMT,
              a.COPAY_AMT,
              a.PROVIDER_NAME,
              a.SRVC_ADDR1,
              a.SRVC_CITY,
              a.SRVC_STATE,
              a.SRVC_ZIP,
              a.SRVC_PHONE,
              a.SRVC_COUNTY,
              a.SPECIALTY,
              a.PLACE_OF_SRVC_DESC,
              a.CPT_CODES_DESCRIPTION,
              a.REV_CODE,
              a.REVENUE_CODES_DESCRIPTION,
              a.MEDICARE_FEE_URBAN_RATE,
              a.MEDICARE_FEE_RURAL_RATE

        from   MSA.qualified_claim_facility          a

        where  index(UPCASE(a.PROCEDURE_CODE),UPCASE("&SPECIALTY_CPTCODE")) >
0 and

index(UPCASE(a.SRVC_CITY),UPCASE("&SERVICE_PROVIDER_CITY")) > 0
        ;
    quit;
%end;

%if (&SPECIALTY_CPTCODE NE %STR( ) and &SERVICE_PROVIDER_CITY NE %STR( )) and
(&SERVICE_PROVIDER_NAME NE %STR( ) and &SERVICE_PLACE_DESCRIPTION EQ %STR(
) and
&SERVICE_PROCEDURE_DESCRIPTION EQ %STR( ))
%then
%do;
    %let t1 = %str(Summary of CPT CODE by Specific Provider Name and City
Report);
    proc sql;
        create table specialty_tbl as
        select a.ALLOWED_AMOUNT,
              a.COINS_AMT,
              a.COPAY_AMT,
              a.PROVIDER_NAME,
              a.SRVC_ADDR1,
              a.SRVC_CITY,
              a.SRVC_STATE,
              a.SRVC_ZIP,

```

Medicare MSA Transparency Tool

```

a.SRVC_PHONE,
  a.SRVC_COUNTY,
a.SPECIALTY,
  a.PLACE_OF_SRVC_DESC,
a.CPT_CODES_DESCRIPTION,
  a.REV_CODE,
a.REVENUE_CODES_DESCRIPTION,
a.MEDICARE_FEE_URBAN_RATE,
a.MEDICARE_FEE_RURAL_RATE

from MSA.qualified_claim_facility a

where index(UPCASE(a.PROCEDURE_CODE),UPCASE("&SPECIALTY_CPTCODE")) >
0 and
index(UPCASE(a.SRVC_CITY),UPCASE("&SERVICE_PROVIDER_CITY")) > 0 and
index(UPCASE(a.PROVIDER_NAME),UPCASE("&SERVICE_PROVIDER_NAME")) > 0
;
quit;
%end;

%if (&SPECIALTY_CPTCODE NE %STR( ) and &SERVICE_PROVIDER_CITY NE %STR( ))
and
(&SERVICE_PROVIDER_NAME NE %STR( ) and &SERVICE_PLACE_DESCRIPTION NE %STR(
) and
&SERVICE_PROCEDURE_DESCRIPTION EQ %STR( ))
%then
%do;
  %let t1 = %str(Summary of CPT CODE by Specific Provider Name, Place of
Service and City Report);
  proc sql;
    create table specialty_tbl as
    select a.ALLOWED_AMOUNT,
      a.COINS_AMT,
      a.COPAY_AMT,
      a.PROVIDER_NAME,
      a.SRVC_ADDR1,
      a.SRVC_CITY,
      a.SRVC_STATE,
      a.SRVC_ZIP,
      a.SRVC_PHONE,
      a.SRVC_COUNTY,
      a.SPECIALTY,
      a.PLACE_OF_SRVC_DESC,
      a.CPT_CODES_DESCRIPTION,
      a.REV_CODE,
      a.REVENUE_CODES_DESCRIPTION,
      a.MEDICARE_FEE_URBAN_RATE,
      a.MEDICARE_FEE_RURAL_RATE

from MSA.qualified_claim_facility a

```


Medicare MSA Transparency Tool

```

0      where  index(UPCASE(a.PROCEDURE_CODE),UPCASE("&SPECIALTY_CPTCODE")) >
and

index(UPCASE(a.SRVC_CITY),UPCASE("&SERVICE_PROVIDER_CITY")) > 0      and

index(UPCASE(a.PROVIDER_NAME),UPCASE("&SERVICE_PROVIDER_NAME")) > 0 and

index(UPCASE(a.PLACE_OF_SRVC_DESC),UPCASE("&SERVICE_PLACE_DESCRIPTION")) > 0
;
quit;
%end;

%if (&SPECIALTY_CPTCODE NE %STR( ) and &SERVICE_PROVIDER_CITY NE %STR( )) and
(&SERVICE_PROVIDER_NAME NE %STR( ) and &SERVICE_PLACE_DESCRIPTION NE %STR(
) and
&SERVICE_PROCEDURE_DESCRIPTION NE %STR( ))
%then
%do;
    %let t1 = %str(Summary of CPT CODE by Specific Provider Name, Place of
Service, Procedure Description and City Report);
    proc sql;
        create table specialty_tbl as
        select a.ALLOWED_AMOUNT,
              a.COINS_AMT,
              a.COPAY_AMT,
              a.PROVIDER_NAME,
              a.SRVC_ADDR1,
              a.SRVC_CITY,
              a.SRVC_STATE,
              a.SRVC_ZIP,
              a.SRVC_PHONE,
              a.SRVC_COUNTY,
              a.SPECIALTY,
              a.PLACE_OF_SRVC_DESC,
              a.CPT_CODES_DESCRIPTION,
              a.REV_CODE,
              a.REVENUE_CODES_DESCRIPTION,
              a.MEDICARE_FEE_URBAN_RATE,
              a.MEDICARE_FEE_RURAL_RATE

        from  MSA.qualified_claim_facility          a

        where  index(UPCASE(a.PROCEDURE_CODE),UPCASE("&SPECIALTY_CPTCODE")) >
0      and

index(UPCASE(a.SRVC_CITY),UPCASE("&SERVICE_PROVIDER_CITY")) > 0
and

index(UPCASE(a.PROVIDER_NAME),UPCASE("&SERVICE_PROVIDER_NAME")) > 0
and

index(UPCASE(a.PLACE_OF_SRVC_DESC),UPCASE("&SERVICE_PLACE_DESCRIPTION")) > 0
and

```

Medicare MSA Transparency Tool

```
index(UPCASE(a.PROCEDURE_DESC),UPCASE("&SERVICE_PROCEDURE_DESCRIPTION")) > 0
;
quit;
%end;

%mend process;

%process;

proc sort data=specialty_tbl out=specialty_out noduprecs;
  by _all_;
run;

data specialty_rpt;
  set specialty_out;
run;

options nocenter ps=100;
proc report data=specialty_rpt headline split='*' missing spacing=0;

title1 'Network Health';
title2 &T1;

column SRVC_ZIP PROVIDER_NAME SRVC_CITY SRVC_ADDR1 SRVC_COUNTY SRVC_STATE
SRVC_PHONE SPECIALTY
        PLACE_OF_SRVC_DESC CPT_CODES_DESCRIPTION REV_CODE
REVENUE_CODES_DESCRIPTION MEDICARE_FEE_URBAN_RATE
        MEDICARE_FEE_RURAL_RATE ALLOWED_AMOUNT COINS_AMT COPAY_AMT ;

define srvc_zip / order order=data 'Service*Zip' width=10;
define provider_name / group 'Provider*Name' width=12;
define srvc_city / group 'Service*City' width=11;
define srvc_addr1 / group 'Service*Address' width=15;
define srvc_county / group 'Service*County' width=13;
define srvc_state / group 'Service*State' width=12;
define srvc_phone / group 'Service*phone' width=12;
define specialty / group 'Specialty' width=9;
define place_of_srvc_desc / group 'Place_of*Service_Description' width=28;
define cpt_codes_description / group 'CPT_CODE*Description' width=19;
define rev_code / group 'REV*CODE' width=7;
define revenue_codes_description / display 'REV_CODE*Description' width=19;
define medicare_fee_urban_rate / display 'Medicare_Fee*Urban_Rate' width=22;
define medicare_fee_rural_rate / display 'Medicare_Fee*Rural_Rate' width=22;
define allowed_amount / display 'Allowed*Amount' width=14;
define coins_amt / display 'Coins*Amount' width=12;
define copay_amt / display 'Copay*Amount' width=12;

run;

%STPEND;
```

Medicare MSA Transparency Tool

Note: Speciality CPT Code Description program uses as its parm input a specific Cpt Code Description and the 1st four letters of a specific city, ie. 'App' for 'Appleton' in which the specialist resides. The program can be used to provide a subscriber with the name, address, phone, cpt code, cpt code description, revenue code, revenue code description, MSA fee, coinsurance, copayment, and allowed amounts for providers. The program can be modified should additional information, i.e such as a report per 'revenue code' rather than 'cpt code' be required.

Specialty CPT Description Report

```
*ProcessBody;
```

```
%STPBEGIN;
```

```
%global SERVICE_RC_DESCRIPT SERVICE_PROVIDER_NAME SERVICE_PLACE_DESCRIPTION  
SERVICE_PROCEDURE_DESCRIPT ;
```

```
/*-----*/  
/* Timer Start */  
/*-----*/
```

```
data _null_;  
    time = time();  
    call symput('Start',time);  
run;
```

```
options validvarname=v7;
```

```
/*-----*/  
/* Library file Definitions */  
/*-----*/
```

```
%let DRIVE_I=\\affsanfe01\Aff_Departments\Departments_Network_Health; * I  
Drive;  
%let DRIVE_Y=\\affsanfe01\Share_Team; * Y Drive;  
%let DRIVE_G=\\affsassoc00p01;*SAS server;
```

```
/* Global Variable Definitions */
```

```
%global DRIVE_I DRIVE_Y DRIVE_G T1;
```

```
libname star "&DRIVE_G\sasdata\SAS Warehouse\Star";
```

```
libname MSA "&DRIVE_I.\DecisionSupport\vtaylor\MSA Transparency Tool";
```

```
proc datasets library=work memtype=data kill nolist;  
run;
```

```
%macro process;
```

```
/* Select Specialty that Sound like */
```

```
%if (&SERVICE_PROCEDURE_DESCRIPT NE %STR( ) or &SERVICE_RC_DESCRIPT NE %STR( ))  
or  
    (&SERVICE_PROVIDER_NAME EQ %STR( ) or &SERVICE_PLACE_DESCRIPTION EQ %STR(  
))
```

Medicare MSA Transparency Tool

```

%then
%do;
    %let t1 = %str(Summary of CPT Code Description by Specific City Report);
    proc sql;
        create table specialty_tbl as
        select a.ALLOWED_AMOUNT,
              a.COINS_AMT,
              a.COPAY_AMT,
              a.PROVIDER_NAME,
              a.SRVC_ADDR1,
              a.SRVC_CITY,
              a.SRVC_STATE,
              a.SRVC_ZIP,
              a.SRVC_PHONE,
              a.SRVC_COUNTY,
              a.SPECIALTY,
              a.PLACE_OF_SRVC_DESC,
              a.CPT_CODES_DESCRIPTION,
              a.REV_CODE,
              a.REVENUE_CODES_DESCRIPTION,
              a.MEDICARE_FEE_URBAN_RATE,
              a.MEDICARE_FEE_RURAL_RATE

        from   MSA.qualified_claim_facility          a

        where
        index(UPCASE(a.PROCEDURE_DESC),UPCASE("&SERVICE_PROCEDURE_DESCRIPTOR")) > 0 or

        index(UPCASE(a.REVENUE_CODES_DESCRIPTION),UPCASE("&SERVICE_RC_DESCRIPTOR")) > 0
        or

        index(UPCASE(a.PROVIDER_NAME),UPCASE("&SERVICE_PROVIDER_NAME")) > 0 or

        index(UPCASE(a.PLACE_OF_SRVC_DESC),UPCASE("&SERVICE_PLACE_DESCRIPTION")) > 0
        ;
        quit;
    %end;

%mend process;

%process;

proc sort data=specialty_tbl out=specialty_out noduprecs;
    by _all_;
run;

data specialty_rpt;
    set specialty_out;
run;

options nocenter ps=100;
proc report data=specialty_rpt headline split='*' missing spacing=0;

title1 'Network Health';

```

Medicare MSA Transparency Tool

```
title2 &T1;

column SRVC_CITY PROVIDER_NAME SRVC_ZIP SRVC_ADDR1 SRVC_COUNTY SRVC_STATE
SRVC_PHONE SPECIALTY
        PLACE_OF_SRVC_DESC CPT_CODES_DESCRIPTION REV_CODE
REVENUE_CODES_DESCRIPTION MEDICARE_FEE_URBAN_RATE
        MEDICARE_FEE_RURAL_RATE ;

define srvc_city / order 'Service*City' width=11;
define provider_name / order 'Provider*Name' width=12;
define srvc_zip / group 'Service*Zip' width=10;
define srvc_addr1 / group 'Service*Address' width=15;
define srvc_county / group 'Service*County' width=13;
define srvc_state / group 'Service*State' width=12;
define srvc_phone / group 'Service*phone' width=12;
define specialty / group 'Specialty' width=9;
define place_of_srvc_desc / group 'Place_of*Service_Description' width=28;
define cpt_codes_description / display 'CPT_CODE*Description' width=19;
define rev_code / display 'REV*CODE' width=7;
define revenue_codes_description / display 'REV_CODE*Description' width=19;
define medicare_fee_urban_rate / group 'Medicare_Fee*Urban_Rate' width=22;
define medicare_fee_rural_rate / group 'Medicare_Fee*Rural_Rate' width=22;
run;

%STPEND;
```

Medicare MSA Transparency Tool

TEST CASES

Summary of Member Provider History Report Mem# 122020008 (split in selections)

Note: Investigate why cpt_code 82947 is missing

Summary of Specialty by Specific City Hospital, Appl, St Elizabeth, Outpatient, Chest

Note: Add Rev_code to report to ensure all are printing.

Summary of Specialty by Specific County Hospital, Brown, St Vincent, Outpatient, head

Note: Add Rev_code to report to ensure all are printing.

Summary of Specialty by Specific County Hospital, Brown, Bellin Memorial, Outpatient, mammogram

Summary of Specialty by Specific CPT cpt_code=74178, Green Bay

Note: Add Rev_code to report to ensure all are printing.

Summary of CPT Code Description by City Waupaca, DM SLF MGMT, Riverside Medical Center, Outpatient

Note: Add Rev_code to report to ensure all are printing.

Medicare MSA Transparency Tool

QNT - Claim Processing TRIZETT

Welcome Vtaylor, You are logged on to the environment: Environment 1 QNT 5.10.003 production Log

[QNT Home](#) | [Claim Processing Home](#) | [Refresh](#) | [External Links](#) | [About](#) | [Help](#)

Claim Number Search

[Referral](#) | [History](#) | [Provider Info](#) | [Member Info](#) | [Required Docs](#) | [Privacy Policy](#) | [Claim Image](#) | [Ext XML](#) | [TTS](#) | [Process Details](#) | [HRA Info](#) | [Rebill Carrier](#) | [Claim Audit](#) | [Reserved Est](#)

Claims > Service No Memos / Alerts Found

Claims: **140140177600** Provider: **HOLY FAMILY MEMORIAL MEDICAL CENTER**

Member/Secondary ID: **MILLER, SANDRA K/M00676618** DOS: **01/07/2014 - 01/07/2014**

[Delete](#) [Referral](#)

Select	Claimline	Code	Description	HCPCS / NDC	Approved	Modifier	Service Date
<input type="radio"/>	1	0320	Radiology - Diagnostic-General Classification	77080		59	1/7/2014
<input type="radio"/>	2	0320	Radiology - Diagnostic-General Classification	77081		GZ	1/7/2014
<input type="radio"/>	3	0403	Other Imaging Services-Screening Mammography	77052			1/7/2014
<input type="radio"/>	4	0403	Other Imaging Services-Screening Mammography	G0202			1/7/2014
<input type="radio"/>	5						

Claim Header Total: \$1,041.30

Units: 4

Claim Line Total: \$1,041.30

Disallowed: \$853.81

[Next Step](#) [Save](#)

Copyright © 2014 QNT Corporation

	HCPCS / NDC	Approved	Modifier	Svc Date	Billed Units	Charge	
	77080		59	1/7/2014	1	\$505.30	\$
	77081		GZ	1/7/2014	1	\$187.10	\$1
hy	77052			1/7/2014	1	\$74.10	
hy	G0202			1/7/2014	1	\$275.00	