

CHIA Enrollment Trends
(February 2016 Edition)
MA APCD Programming Code¹

As described in the CHIA Enrollment Trends Technical Appendix, Enrollment Trends monitors unique Massachusetts residents with primary medical insurance coverage using eligibility data, where available and able to be verified, from the Massachusetts All Payer Claims Database (MA APCD). MA APCD membership reports are generated twice per year and sourced from payers' most recently submitted Member Eligibility (ME) files. Up to twenty-four months of membership data are used and segmented by product enrollment dates.

Provided below is the programming code used by CHIA to create payers' MA APCD membership reports for Enrollment Trends. The code was developed with and approved by the top 14 commercial payers (and affiliated entities) but are subject to further revision. CHIA may request supplemental data from payers to augment or replace MA APCD enrollment counts where data does not meet quality standards.

I. Methodology

Enrollment is determined if the last day of the quarter (i.e., Snapshot Date) is within the Product Enrollment Start Date (ME041) and Product Enrollment End Date (ME042) of a member record (i.e., $ME041 \leq \text{Snapshot Date} \leq ME042$). A member record with NULL values for either ME041 or ME042 is considered actively enrolled. Members are distinguished using data element HashCarrierSpecificUniqueMemberID (ME107) by OrgID (ME001).

MA APCD membership reports do not count duplicate member records. A unique member record is selected according to the following criteria, in order of significance:

Subset by the following:

1. Massachusetts resident (ME016=MA)
2. Medical coverage (ME018=1)
3. Primary insurance² (ME028=1)

De-duplicate by the following order:

1. Coverage type³ (ME029)
 - a. ME029=(ASO, ASW) for Self-Insured
 - b. ME029=(UND) for Fully-Insured
 - c. ME029=(STN, OTH) for Other
2. Last activity date (ME056)
3. Member Eligibility ID (Derived-ME05)

II. Programming Code

MA APCD data extraction and aggregation are performed in SAS, using SQL pass-through to access data in a Netezza environment.

a. SAS/SQL Code

```
/*  
SAS/SQL CODE TO EXTRACT AGGREGATE APCD MEMBERSHIP DATA FOR ENROLLMENT TRENDS REPORTING  
*/
```

¹ CHIA is providing this programming code as a convenience. It has been prepared for informational purposes only and is based on information believed to be reliable. The programming code is subject to change without notice. CHIA does not provide any guarantee or opinion on its accuracy. CHIA disclaims any liability for the improper or incorrect use of the information contained herein.

² Include ME028=(3, 4) if the payer stores primary insurance as such in its system.

³ This selection hierarchy is to ensure that self-insured plans are given preference over fully-insured plans for members with concurrent enrollment in both self-insured and fully-insured coverage.

```

*----- BEGIN ACTION ITEMS -----

1. Initialize global variables
2. Perform additional action items below prior to running program

----- END ACTION ITEMS -----;

* Initialize global variables;
%let DataDir= C:\EnrollmentTrends\output;
%let SYM= 201509;
%let OrgIdList= (290, 291, 295, 296, 300, 301, 302, 312, 3505, 3735, 4962, 7041, 7655, 7789, 8026, 8647, 10353, 10441,
10444, 10632, 10647, 10920, 10926, 10929, 11474, 11726, 12122, 12226);
%let Snapshot1= '2013-12-31'; *First snapshot date, format: 'yyyy-mm-dd';
%let nMonths= 3; *# of months to increment snapshot dates, Default=3 (e.g. nMonths=1 means monthly, nMonths=3
means quarterly);
%let nTimeSpan= 8; *# of time spans, Default=8 (e.g. if n=3 and nTimeSpan=8, then program will generate a snapshot
date every 3 months, i.e. every quarter, for 8 quarters);

* Fixed global variables;
libname out "&DataDir.";
%let OutFile= out.ME_ET_&SYM.;
%let StartYear= %substr(&snapshot1,2,4); *Year of first snapshot date;
%let StartMonth= %substr(&snapshot1,7,2); *Month of first snapshot date;
%let StartDay= %substr(&snapshot1,10,2); *Day of first snapshot date;

option compress=binary;

/*****
Begin: Netezza summary data extract
*****/

*----- BEGIN ACTION ITEMS -----

Perform the checks below prior to running code

1. Review "Processing Summary" below for previous processing info
2. Check that source data is coming from correct data source

----- END ACTION ITEMS -----;

* Processing Summary:

As of 11/17/2015: Extract 8 quarters, starting with snapshot date of 12/31/2013,
of ME data for ET. The below dataset has the following characteristics:

1) The dataset has been de-duped (to_keep_v1= 1)
2) The dataset contains only OrgIDs relevant to Enrollment Trends
3) The dataset contains two Product fields: ProductID and ProductLineofBusinessModel
4) The dataset contains demographic fields: AgeGroup, County, and MemberGender
;

proc sql;
connect to odbc (dsn=odbc_database);
create table &OutFile. as
select * from connection to odbc (
with DateList as (select add_months(cast(&startYear.&startMonth.&startDay. as datetime), &nMonths.*idx) as dates
from (select &Snapshot1. :: date start_date) starting_date
cross join _v_vector_idx
where idx < &nTimeSpan.
)
select OrgID
, SubmissionControlID
, SubmissionYearMonth
, Dates
, InsuranceTypeCodeProduct
, MedicalCoverage
, PrimaryInsuranceIndicator
, CoverageType
, MarketCategoryCode
, SpecialCoverage
, ProductID
, ProductLineOfBusinessModel
, RACP
, MemberMA
, County
, MemberGender
, AgeGroup

```

```

, PurchasedThroughMassachusettsExchangeFlag
, count(*) as nrecs
from (
select MemberEligibilityID
, me.OrgID
, me.SubmissionControlID
, me.SubmissionYearMonth
, InsuranceTypeCodeProduct
, MedicalCoverage
, PrimaryInsuranceIndicator
, CoverageType
, MarketCategoryCode
, SpecialCoverage
, cast(ProductIDNumber_Linking_ID as char(20)) as ProductID
, prd.ProductLineOfBusinessModel
, ProductEnrollmentStartDate as StartDate
, isnull(ProductEnrollmentEndDate,cast('99991231' as datetime)) as EndDate
, LastActivityDate
, HashCarrierSpecificUniqueMemberID as UniqueMemberID
, MemberGenderCleaned as MemberGender
, case when months_between(Dates,ad.DT)/12 >= 0 and months_between(Dates,ad.DT)/12 < 1 then '< 1'
when months_between(Dates,ad.DT)/12 >= 1 and months_between(Dates,ad.DT)/12 < 10 then '1-9'
when months_between(Dates,ad.DT)/12 >= 10 and months_between(Dates,ad.DT)/12 < 20 then '10-19'
when months_between(Dates,ad.DT)/12 >= 20 and months_between(Dates,ad.DT)/12 < 27 then '20-26'
when months_between(Dates,ad.DT)/12 >= 27 and months_between(Dates,ad.DT)/12 < 45 then '27-44'
when months_between(Dates,ad.DT)/12 >= 45 and months_between(Dates,ad.DT)/12 < 65 then '45-64'
when months_between(Dates,ad.DT)/12 >= 65 and months_between(Dates,ad.DT)/12 < 75 then '65-74'
when months_between(Dates,ad.DT)/12 >= 75 and months_between(Dates,ad.DT)/12 < 85 then '75-84'
when months_between(Dates,ad.DT)/12 >= 85 and months_between(Dates,ad.DT)/12 < 121 then '85+'
else 'Unknown'
end as AgeGroup
, RiskAdjustmentCoveredBenefitPlan as RACP
, case when Standardized_MemberStateorProvince = 'MA' then 1 else 0 end as MemberMA
, case when Standardized_MemberStateorProvince = 'MA' then Standardized_MemberCounty else 'Other' end as County
, PurchasedThroughMassachusettsExchangeFlag
, Dates
, case when lead(me.MemberEligibilityID) over (partition by me.OrgID, HashCarrierSpecificUniqueMemberID, Dates
order by case when Standardized_MemberStateorProvince='MA' then 1 else 0 end
, case when MedicalCoverage='1' then 1 else 0 end
, case when PrimaryInsuranceIndicator in ('1','3','4') then 1 else 0 end
, case when CoverageType in ('ASO', 'ASW') then 1
when CoverageType='UND' then 0
else -1 end
, LastActivityDate
, me.MemberEligibilityID)
is null then 1 else 0
end as to_keep_v1 /* De-duplication hierarchy */
FROM APCD_MemberEligibility as me
inner join Datelist
on Dates between ProductEnrollmentStartDate and isnull(ProductEnrollmentEndDate,cast('99991231' as
datetime))
left join APCD_Product as prd
on me.OrgID=prd.OrgID and me.ProductIDNumber_Linking_ID=prd.LinkingProductID and
prd.LinkingProductDelegate=1
left join DATE_TABLE as ad /* Join to a date table to process only valid Member Date of Birth values */
on me.MemberDateofBirth = ad.DT_YYYYMMDD
where me.SubmissionYearMonth=&SYM.
and me.OrgID in &OrgIdList.
) a
where to_keep_v1=1 /* Save out only de-duplicated records */
group by OrgID
, SubmissionControlID
, SubmissionYearMonth
, Dates
, InsuranceTypeCodeProduct
, MedicalCoverage
, PrimaryInsuranceIndicator
, CoverageType
, MarketCategoryCode
, SpecialCoverage
, ProductID
, ProductLineOfBusinessModel
, RACP
, MemberMA
, County
, MemberGender
, AgeGroup
, PurchasedThroughMassachusettsExchangeFlag
order by 1,2,3,4
);
disconnect from odbc;

```

```

quit;

/*****
End: Summary data extract
*****/

/*****
Begin: Export summary data extract
*****/

*   Part 1 of 2: Save out de-duped enrollment counts with ET standard fields
The below dataset has the following characteristics:
1) The dataset has been de-duped (to_keep_v1= 1)
2) The dataset contains only OrgIDs relevant to Enrollment Trends
3) The dataset contains two Product fields: ProductID and ProductLineofBusinessModel
4) The dataset DOES NOT contain demographic fields: AgeGroup, County, and MemberGender
;

proc sql;
create table out.ET_standard as
select OrgID
, SubmissionControlID
, SubmissionYearMonth
, datepart(Dates) as Dates format MDDYY10.
, InsuranceTypeCodeProduct
, MedicalCoverage
, PrimaryInsuranceIndicator
, CoverageType
, MarketCategoryCode
, SpecialCoverage
, ProductID
, ProductLineOfBusinessModel
, RACP
, MemberMA
, PurchasedThroughMassachusettsExc
/* , County*/
/* , MemberGender*/
/* , AgeGroup*/
, sum(nrecs) as Member_Count
from out.ME_ET_201509
group by OrgID
, SubmissionControlID
, SubmissionYearMonth
, calculated Dates
, InsuranceTypeCodeProduct
, MedicalCoverage
, PrimaryInsuranceIndicator
, CoverageType
, MarketCategoryCode
, SpecialCoverage
, ProductID
, ProductLineOfBusinessModel
, RACP
, MemberMA
, PurchasedThroughMassachusettsExc
/* , County*/
/* , MemberGender*/
/* , AgeGroup*/
;
quit;

*   Part 2 of 2: Save out de-duped enrollment counts for
ET standard fields plus demographic fields (age group/county/gender)
for ET population only, due to file size limitations.
The below dataset has the following characteristics:
1) The dataset has been de-duped (to_keep_v1= 1)
2) The dataset contains only OrgIDs relevant to Enrollment Trends
3) The dataset DOES NOT contain two Product fields: ProductID and ProductLineofBusinessModel
4) The dataset contains demographic fields: AgeGroup, County, and MemberGender
5) The dataset has been subsetted for the ET population (MA residents with primary medical coverage)
;

proc sql;
create table out.ET_demogs as
select OrgID
, SubmissionControlID

```

```

, SubmissionYearMonth
, datepart(Dates) as Dates format MMDYY10.
, InsuranceTypeCodeProduct
, MedicalCoverage
, PrimaryInsuranceIndicator
, CoverageType
, MarketCategoryCode
, SpecialCoverage
/* , ProductID*/
/* , ProductLineOfBusinessModel*/
, RACP
, MemberMA
, PurchasedThroughMassachusettsExc
, County
, MemberGender
, AgeGroup
, sum(nrecs) as Member_Count
from out.ME_ET_201509
where MemberMA= 1 and PrimaryInsuranceIndicator in ('1', '3', '4') and MedicalCoverage= '1'
group by OrgID
, SubmissionControlID
, SubmissionYearMonth
, calculated Dates
, InsuranceTypeCodeProduct
, MedicalCoverage
, PrimaryInsuranceIndicator
, CoverageType
, MarketCategoryCode
, SpecialCoverage
/* , ProductID*/
/* , ProductLineOfBusinessModel*/
, RACP
, MemberMA
, PurchasedThroughMassachusettsExc
, County
, MemberGender
, AgeGroup
;
quit;

```

* Apply hard-coded groupings and export dataset(s) to Excel
Macro Input: dataEdit(name of input dataset, name of output dataset, name to identify dataset for Excel filename)
Macro Output: a temporary SAS dataset and an Excel xlsx-file

```

%macro dataEdit(dataInfile, dataOutfile, dataName);
PROC SQL;
CREATE TABLE &dataOutfile. AS
SELECT *
, CASE WHEN MarketCategoryCode in ('IND', 'GCV', 'ISCO') THEN 'Individual'
WHEN MarketCategoryCode in ('GS1', 'GS2', 'GS3') THEN 'Small Group (1-25)'
WHEN MarketCategoryCode='GS4' THEN 'Small Group (26-50)'
WHEN MarketCategoryCode='GLG1' THEN 'Mid-Size Group (51-100)'
WHEN MarketCategoryCode in ('GLG2', 'GLG3') THEN 'Large Group (101-500)'
WHEN MarketCategoryCode='GLG4' THEN 'Jumbo-Size Group (501+)'
WHEN MarketCategoryCode='GSA' THEN 'Qualified Association'
ELSE 'Other'
END AS GROUPSIZE
/* - Finding: Tufts' (OrgID=8647) self-insured members are classified as CoverageType='OTH'
in APCD.
- Resolution: Need to hardcode as MARKET='Self-Insured' for ET reporting. */
, CASE WHEN ORGID=8647 THEN
CASE WHEN CoverageType in ('OTH', 'ASO', 'ASW') THEN 'Self-Insured'
WHEN CoverageType='UND' THEN 'Fully-Insured'
ELSE 'Other' END
/* - Finding: Fallon (OrgID=296) has a fully-insured line of business, but members
are classified as CoverageType='OTH'.
- Resolution: Need to hardcode as MARKET='Fully-Insured' for ET reporting. */
WHEN ORGID=296 THEN 'Fully-Insured'
/* - Finding: Fallon (OrgID=8026) classified its self-insured members as
InsuranceTypeCodeProduct='09'. As a result, cannot break out its self-
insured population by product type. All other members are fully-insured.
- Resolution: Need to hardcode as MARKET='Self-Insured' for InsuranceTypeCodeProduct='09'
and MARKET='Fully-Insured' for all other members for ET reporting. */
WHEN ORGID=8026 THEN
CASE WHEN InsuranceTypeCodeProduct='09' THEN 'Self-Insured'
ELSE 'Fully-Insured' END
/* Finding: United (OrgID=7789) is a student health plan, with members classified as
CoverageType='STN'.
Resolutions: Need to hardcode as MARKET='Fully-Insured' for ET reporting. */

```

```

WHEN ORGID=7789 THEN
  CASE WHEN CoverageType in ('STN', 'UND') THEN 'Fully-Insured'
        WHEN CoverageType in ('ASO', 'ASW') THEN 'Self-Insured'
        ELSE 'Other' END
ELSE CASE WHEN CoverageType='UND' THEN 'Fully-Insured'
        WHEN CoverageType in ('ASO', 'ASW') THEN 'Self-Insured'
        ELSE 'Other' END
END AS MARKET
/* - Finding: Aetna (OrgID=10929) is a Medicare plan, but members are classified
   as various Product Types ('12', 'HM', 'MD').
   - Resolution: Need to hardcode as PLANTYPE='Medicare' for ET reporting. */
, CASE WHEN ORGID=10929 THEN 'Medicare'
/* - Finding: Fallon (OrgID=12122) is a OneCare plan,
   but members are classified as Product Type='ZZ'.
   - Resolution: Need to hardcode as PLANTYPE='OneCare' for ET reporting. */
WHEN ORGID=12122 THEN 'OneCare'
/* - Finding: Network Health's (OrgID=4962) MSP population is classified as Product Type='MO'.
   - Resolution: Need to hardcode as PLANTYPE='Medical Security Program' for Product Type='MO'
   for ET reporting. */
WHEN ORGID=4962 THEN
  CASE WHEN InsuranceTypeCodeProduct='MO' THEN 'Medical Security Program'
        WHEN InsuranceTypeCodeProduct='MC' THEN CASE WHEN MarketCategoryCode= 'ISCO' THEN 'OneCare'
ELSE 'Medicaid' END
        WHEN InsuranceTypeCodeProduct='CC' OR SpecialCoverage='CC' THEN 'Commonwealth Care'
        WHEN InsuranceTypeCodeProduct='CE' THEN 'Commonwealth Choice'
        WHEN InsuranceTypeCodeProduct='HM' THEN 'HMO'
        WHEN InsuranceTypeCodeProduct='12' THEN 'PPO'
        WHEN InsuranceTypeCodeProduct='13' THEN 'POS'
        WHEN InsuranceTypeCodeProduct='14' THEN 'EPO'
        WHEN InsuranceTypeCodeProduct='15' THEN 'Indemnity'
        WHEN InsuranceTypeCodeProduct in ('16', 'HN', '20') THEN 'Medicare'
        WHEN InsuranceTypeCodeProduct='SC' THEN 'Senior Care Options'
        ELSE 'Other' END
/* - Finding: Some of Tufts' (OrgID=8647) PPO GIC members transitioned to a POS plan as of July 1, 2015;
   these members were identified as Product Type='ZZ' instead of Product Type='13'
   - Resolution: Need to hardcode as PLANTYPE='POS' for Product Type='ZZ' for ET reporting. */
WHEN ORGID=8647 THEN
  CASE WHEN InsuranceTypeCodeProduct in ('MC', 'MO') THEN 'Medicaid'
        WHEN InsuranceTypeCodeProduct='CC' OR SpecialCoverage='CC' THEN 'Commonwealth Care'
        WHEN InsuranceTypeCodeProduct='CE' THEN 'Commonwealth Choice'
        WHEN InsuranceTypeCodeProduct='HM' THEN 'HMO'
        WHEN InsuranceTypeCodeProduct='12' THEN 'PPO'
        WHEN InsuranceTypeCodeProduct in ('13', 'ZZ') THEN 'POS'
        WHEN InsuranceTypeCodeProduct='14' THEN 'EPO'
        WHEN InsuranceTypeCodeProduct='15' THEN 'Indemnity'
        WHEN InsuranceTypeCodeProduct in ('16', 'HN', '20') THEN 'Medicare'
        WHEN InsuranceTypeCodeProduct='SC' THEN 'Senior Care Options'
        ELSE 'Other' END
ELSE CASE WHEN InsuranceTypeCodeProduct in ('MC', 'MO') THEN 'Medicaid'
        WHEN InsuranceTypeCodeProduct='CC' OR SpecialCoverage='CC' THEN 'Commonwealth Care'
        WHEN InsuranceTypeCodeProduct='CE' THEN 'Commonwealth Choice'
        WHEN InsuranceTypeCodeProduct='HM' THEN 'HMO'
        WHEN InsuranceTypeCodeProduct='12' THEN 'PPO'
        WHEN InsuranceTypeCodeProduct='13' THEN 'POS'
        WHEN InsuranceTypeCodeProduct='14' THEN 'EPO'
        WHEN InsuranceTypeCodeProduct='15' THEN 'Indemnity'
        WHEN InsuranceTypeCodeProduct in ('16', 'HN', '20') THEN 'Medicare'
        WHEN InsuranceTypeCodeProduct='SC' THEN 'Senior Care Options'
        ELSE 'Other' END
END AS PLANTYPE
, CASE WHEN OrgID in (290, 7655, 10353, 10441, 10442, 10647, 10929, 11745) THEN 'Aetna'
  WHEN OrgID in (10632) THEN 'Anthem'
  WHEN OrgID in (291) THEN 'BCBS'
  WHEN OrgID in (3505) THEN 'BMC'
  WHEN OrgID in (10920) THEN 'CeltiCare'
  WHEN OrgID in (293, 295, 7422, 10447, 11474, 11499, 11726, 11215) THEN 'CIGNA'
  WHEN OrgID in (7041) THEN 'ConnectiCare'
  WHEN OrgID in (296, 8026, 12122) THEN 'Fallon'
  WHEN OrgID in (300) THEN 'HPHC'
  WHEN OrgID in (302) THEN 'HPI' /* In later processing, re-classify HPI OrgID=302 as CompanyName=HPHC */
  WHEN OrgID in (301) THEN 'HNE'
  WHEN OrgID in (3156) THEN 'MassHealth'
  WHEN OrgID in (12226) THEN 'Minuteman'
  WHEN OrgID in (3735) THEN 'NHP'
  WHEN OrgID in (4962) THEN 'NetworkHealth' /* In later processing, re-classify Network Health OrgID=4962 as
CompanyName=Tufts */
  WHEN OrgID in (8647) THEN 'Tufts'
  WHEN OrgID in (312, 313, 7789, 10444, 10925, 10926, 10932, 10933, 10934, 10935) THEN 'United'
  ELSE 'Other'
END AS CompanyName
FROM &dataInfile.;

```

```

QUIT;

* Double check to make sure the County field is a true Massachusetts-based county ;
%IF &dataName.=Demogs %THEN %DO;
  PROC SQL;
    CREATE TABLE &dataOutfile. AS
    SELECT *
      , CASE WHEN UPCASE(County) in ('BARNSTABLE', 'BERKSHIRE', 'BRISTOL', 'DUKES', 'ESSEX',
        'FRANKLIN', 'HAMPDEN', 'HAMPSHIRE', 'MIDDLESEX', 'NANTUCKET', 'NORFOLK',
        'PLYMOUTH', 'SUFFOLK', 'WORCESTER', '') THEN County
        ELSE 'Other'
      END AS MEMBERCOUNTY
    FROM &dataOutfile.
  QUIT;
%END;

/* Export data set to Excel file */
PROC EXPORT DATA= &dataOutfile.
  OUTFILE="&DataDir.\ET_&dataName._&SYM..xlsx"
  DBMS=xlsx
  REPLACE;
  RUN;

%mend dataEdit;

%dataEdit(out.ET_Standard, out.report_Standard, Standard);
%dataEdit(out.ET_Demogs, out.report_Demogs, Demogs);

/*****
End: Export summary data extract
*****/

```