**17 March 2016**

DMHRSi HR Basic File

for the

MHS Data Repository (MDR)

(Version 1.2.0)

Future Specification

**Revision History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Para/Tbl/Fig** | **Originator** | **Description of Change** |
| 1.0.0 | 04/10/2012 | Whole document | H. Escobar | Baseline Document |
| 1.0.1 | 05/02/2012 | Table 1 | H. Escobar | Revised Person\_Type and Service variable lengths |
| 1.1.0 | 04/07/2015 | Sections I, II, III  | H. Escobar | Update ICD reference; remove requirement for annual full data refresh and weekly raw data updates |
| 1.2.0 | 3/17/2016 | Section V, Table 1 | H. Escobar | Added creation of Derived\_DODOCC through lookup by SUOC against EASIV SUOC/DODOCC table. |

**MDR DMHRSi-HR Basic File**

1. Source

Data capture system: Defense Medical Human Resources System (internet) (DMHRSi) documented in **MDR\_DHSS\_ICD\_16Dec2014.docx, dated 16 Dec 2014**.

1. Transmission (Format and Frequency)

The initial file load is a one-time requirement containing selected DMHRSi-HR data from 1 Oct 2008 (FY09 Start) to 30 Sept 2011 (FY11 End). Subsequently, DHSS will provide monthly raw ASCII text “!” delimited DMHRSi HR data extracts via FTP as described in the interface control document (ICD) referenced above. The incremental extracts reflect updated or added DMHRSi personnel records since the most recent extract (see Appendix A for incremental extract SQL code). DHSS staff will provide the following DMHRSi HR extract:

|  |  |
| --- | --- |
| **Table Name** | **Description** |
| DMHRSi-HR Extract(Table A-1: dmhrsi.txt) | Contains incrementally updated DMHRSi-HR raw data since most recent extract |

1. Organization and batching
2. DMHRSi extracts are received and processed monthly.

1. After processing, each updated DMHRSi-HR Basic file is saved as individual SAS datasets containing DMHRSi-HR data from 1 Oct 2008 to extract date of the most recent incremental file.
2. Receiving Filters
3. Processor should scan each incoming raw incremental update and remove non-visible characters imbedded in the raw data observations (i.e., tabs, line breaks, etc.).
4. Filter and correct erroneous delimiter characters “!” imbedded inside valid observations (i.e., "KV!MFV9P" vs. "KV1MFV9P").
5. Processor should remove duplicate records that may be included in the raw files.
6. Field Transformations and Deletions for MDR Core Database
7. The existing master DMHRSi-HR Basic file will be updated monthly with incremental data sets. The Processor should rely on a combination of “Person ID” and “Assignment Effective Start Date” to assess whether person records received in the incremental files are present in the master file. If so, the specific record should be replaced with the content in the incremental DMHRSi-HR extract. If there is no “Person ID” and “Assignment Effective Start Date” match to the existing DMHRSi-HR dataset, the content in the incremental DMHRSi-HR extract should be appended to the master DMHRSi-HR dataset.
8. The Processor will note the processing date personnel records contained in the incremental data input files are added using the PROCDATE field as “MM/DD/YYYY”.
9. All dates should use SAS format “MMDDYY10.”.
10. A new variable “Derived\_DODOCC” will be created in the DMHRSi HR SAS file during monthly processing by merging against the most recent SUOC/DODOCC reference file available by SUOC and SKILL\_TYPE. Presently, this reference file is /mdr/aref/dmhrsi/d160316/occmap.sas7bdat. As map updates occur, newer files will be posted in /mdr/aref/dmhrsi/. Updates will be made at least annually, or as needed. All observations in the source DMHRSi HR SAS dataset should be retained, and only matching observations in the reference file should be kept. In addition, observations in the post-merged DMHRSi HR SAS dataset where Skill\_Type is equal to “5” and SUOC is in the range "0","00","000","0000","00000","0000E","0000O","0000C", the new “Derived\_DODOCC” variable should be set to “NONMED”. Below is sample SAS code for this step where SAS dataset “one” is the DMHRSi HR SAS dataset and SAS dataset “two” is the most recent SUOC/DODOCC reference file:

**proc** **sort** data=one;

by suoc skill\_type; **run**;

**proc** **sort** data=two;

by suoc skill\_type; **run**;

**data** comb;

merge one(in=dat) two;

by suoc skill\_type;

if dat;

if skill\_type ="5" and suoc in ("0","00","000","0000","00000","0000E","0000O","0000C") then derived\_dodocc = "NONMED";

run;

1. File layout and content

The MDR DMHRSi-HR Basic file is a SAS Data Set.

**TABLE 1: MDR DMHRSi-HR Basic FILE**

| **Field Name** | **Field Length** | **Data Type** | **SAS Name** | **Functional Description** |
| --- | --- | --- | --- | --- |
| Processing Date (SAS Date) | 10 | Date | PROCDATE | Date of record processingFormat MM/DD/YYYY |
| SSN | 11 | Char | SSN | Social Security Number |
| EDIPN | 10 | Char | EDIPN | DEERS Electronic Data Interchange Person Number |
| NPI | 10 | Char | NPI | National Provider Identifier |
| Person ID | 8 | Char | PERSON\_ID | Unique DMHRSi Person id |
| Employee Number | 8 | Char | EMP\_NUM | Unique DMHRSi Person Number |
| Employee Assignment DMIS ID | 4 | Char | ASSIG\_DMISID | Employee's DMIS ID of Assignment |
| Employee People Group DMIS ID | 4 | Char | PG\_DMISID | People Group Org DMISID where labor time is recorded |
| Position ID | 8 | Char | POSITION\_ID | Unique DMHRSi Position id |
| Job ID | 8 | Char | JOB\_ID | Unique DMHRSi Job id |
| Organization ID | 8 | Char | ORG\_ID | Unique DMHRSi Organization ID |
| People Group ID | 8 | Char | PG\_ID | Unique DMHRSi People Group ID |
| Assignment Effective Start Date (SAS Date) | 10 | Date | ASSIG\_START | Employee's Assignment Start DateFormat MM/DD/YYYY |
| Assignment Effective End Date (SAS Date) | 10 | Date | ASSIG\_END | Employee's Assignment End DateFormat MM/DD/YYYY |
| Person Effective Start Date (SAS Date) | 10 | Date | PERSON\_START | HR Record Start DateFormat MM/DD/YYYY |
| Person Effective End Date (SAS Date) | 10 | Date | PERSON\_END | HR Record End DateFormat MM/DD/YYYY |
| Original Date of Hire (SAS Date) | 10 | Date | HIRE\_DATE | Employee's Original Date of HireFormat MM/DD/YYYY |
| Assignment DMIS ID Service | 9 | Char | ASSIG\_SERVICE | Branch of Service of Assignment DMISID |
| People Group DMIS ID Service | 9 | Char | PG\_SERVICE | Service Branch of People Group DMISID where time is recorded |
| Last Name | 28 | Char | LASTNAME | Employee last name |
| First Name | 28 | Char | FIRSTNAME | Employee first name |
| Middle Name | 20 | Char | MIDNAME | Employee middle name |
| Suffix | 4 | Char | SUFFIX | Employee Name Suffix |
| Title/Rank | 12 | Char | TITLE\_RANK | Employee's Title or Rank |
| Person Grade | 8 | Char | GRADE | Civilian/Active Duty Grade |
| Person Step | 2 | Char | STEP | Employee's Step associated with grade |
| Gender | 1 | Char | GENDER | Employee Gender |
| Person Type | 22 | Char | PERSON\_TYPE | Employee Personnel Category |
| Skill Type | 1 | Char | SKILL\_TYPE | Person Skill Type |
| Skill Type Suffix | 1 | Char | SKILL\_SUFFIX | Skill Type Suffix Code |
| Organization UIC  | 8 | Char | ORG\_UIC | UIC Associated with employee's organization |
| People Group UIC | 8 | Char | PG\_UIC | UIC Associated with employee's People Group |
| DOD Occupation Code | 10 | Char | DODOCC | DoD Occupation Code associated with job |
| Taxonomy Code 1 | 11 | Char | TAX1 | HIPAA Taxonomy code 1 |
| Taxonomy Code 2 | 11 | Char | TAX2 | HIPAA Taxonomy code 2 |
| Taxonomy Code 3 | 11 | Char | TAX3 | HIPAA Taxonomy code 3 |
| Taxonomy Code 4 | 11 | Char | TAX4 | HIPAA Taxonomy code 4 |
| Taxonomy Code 5 | 11 | Char | TAX5 | HIPAA Taxonomy code 5 |
| Taxonomy Code 6 | 11 | Char | TAX6 | HIPAA Taxonomy code 6 |
| Person Service | 14 | Char | SERVICE | Person's Branch of Service  |
| Person UIC/PAS | 8 | Char | UIC | UIC Associated with employee |
| SUOC | 8 | Char | SUOC | Service-Unique Occupation Code |
| Record ID | 24 | Char | RECORD\_ID | Unique DMHRSi Record Id |
| Creation Date (SAS Date) | 10 | Date | RECORD\_DATE | Date DMHRSi Record was createdFormat MM/DD/YYYY |
| Last Update Date (SAS Date) | 10 | Date | UPDATE\_DATE | Date DMHRSi Record was last updatedFormat MM/DD/YYYY |
| People Group FCC | 4 | Char | PG\_ASSIG\_FCC | Employee's People Group Assignment FCC |
| Derived DoD Occupation Code from EASIV Table | 6 | Char | DERIVED\_DODOCC | Derived DoD Occupation Code from EASIV SUOC to DODOCC Table |

1. Refresh Frequency

The DMHRSi-HR Basic file will be updated monthly with new or changed personnel records since the most recent harvest date.

1. Data Marts

None.

1. Special Outputs

None.

1. Quality assurance

The processor should conduct quality assurance checks on every processed output file to ensure input and output data are valid, complete, and reliable. At a minimum, the processor should:

* Ensure no “Lost Cards” result during raw text import into SAS. This is common if random hex character commands are imbedded in valid observations.
* Compare monthly raw data row counts to ensure changed/added records match raw input row counts.
* Evaluate post-processing values for data that appear out of the ordinary, or not consistent with SME expected values (face validity).

Appendix A: Incremental Extract SQL Code

(Edit Begin/End Incremental Dates as Appropriate)

SET LINESIZE 420

SET SERVEROUTPUT ON SIZE UNLIMITED

SPOOL extract\_output\_mar\_90.txt append

-- first run 82420

-- second run 109995

-- third run 29205

DECLARE

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 /\*\* Version 2 Aug. 10, 2011

 /\*\* Richard Koonce

 /\*\* Modified on Aug. 10, 2011 to add 5 taxonomy columns to the

 /\*\* exiting 1 for a total of 6. These 5 adds are blank place

 /\*\* holder columns only. No additional data is coming from

 /\*\* the database.

 /\*\*

 /\*\* Version 3 Oct. 26, 2011

 /\*\* Modified to add 5 additional columns to report output.

 /\*\* SUOC, Record-ID, Creation-Date, Last-Update-Date and People-Group-FCC.

 /\*\* SUOC is obtained by using a modified copy of existing EAS function.

 /\*\* Record-ID is the rowid from people table.

 /\*\* Creation-Date and Last-Update-Date are from people table.

 /\*\* People-Group-FCC is from the LCA view.

 /\*\* Additional modifications were made to the main cursor to limit

 /\*\* records to prior to Oct-2011 (constant vc\_rec\_change\_date).

 /\*\* Added the dod-person-type view to get FCC.

 /\*\* Version 4

 /\*\* changed vc\_rec\_change\_date to 01-MAR-2012

 /\*\*

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\* Global Constants \*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 vc\_extract\_effect\_date CONSTANT DATE := TO\_DATE('01-OCT-2011'); -- date after which assigments should end

 vc\_rec\_change\_date CONSTANT DATE := TO\_DATE('01-MAR-2012'); -- date after which assignments should not begin.

 vc\_todays\_date CONSTANT DATE := TRUNC(SYSDATE);

 vc\_hr\_max\_end\_date CONSTANT DATE := TO\_DATE('31-DEC-4712'); -- always end of time.

 vc\_ascii\_delimiter CONSTANT VARCHAR2(1) := '!';

 vc\_bulk\_collect\_limit CONSTANT NUMBER := 20000;

 l\_mod CONSTANT NUMBER := 90;

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 CURSOR c\_empl\_info\_recs IS

 SELECT pers.person\_id

 , pers.effective\_start\_date empl\_hr\_start\_date

 , pers.effective\_end\_date empl\_hr\_end\_date

 , ptuse.effective\_start\_date pers\_type\_usage\_start\_dt

 , ptuse.effective\_end\_date pers\_type\_usage\_end\_dt

 , ptuse.object\_version\_number pers\_type\_usage\_ovn

 , pers.national\_identifier

 , REPLACE

 (pers.attribute18,CHR(13)) edipn

 , qual.attribute2 npi

 , pers.employee\_number

 , pers.attribute1 person\_service

 , pers.last\_name

 , pers.first\_name

 , pers.middle\_names

 , pers.suffix

 , pers.title title\_rank

 , pers.sex gender

 , pers.attribute2 grade

 , pers.attribute4 step

 , pers.attribute5 person\_uic

 , pers.start\_date orig\_date\_of\_hire

 , ptyp.user\_person\_type person\_type

 , ptyp.system\_person\_type

 , asgn.position\_id

 , asgn.job\_id

 , asgn.ass\_attribute5 skill\_type

 , asgn.ass\_attribute6 skill\_type\_suffix

 , asgn.effective\_start\_date empl\_assign\_start\_date

 , asgn.effective\_end\_date empl\_assign\_end\_date

 , asgn.organization\_id

 , asgn\_org.attribute13 organization\_uic

 , org\_info.lca\_service organization\_lca\_service

 , org\_info.dmis\_id organization\_lca\_dmis\_id

 , asgn.people\_group\_id

 , group\_info.organization\_id p\_group\_organization\_id

 , group\_info.organization\_uic p\_group\_organization\_uic

 , grp\_lca\_info.lca\_service p\_group\_lca\_service

 , grp\_lca\_info.dmis\_id p\_group\_lca\_dmis\_id

 , per\_eit\_info.pers\_eit\_start\_date

 , per\_eit\_info.pers\_eit\_end\_date

 , per\_eit\_info.dod\_occupation\_code

 , per\_eit\_info.taxonomy\_code

 --- adds Aug-2011

 , pers.rowid person\_rowid

 , pers.creation\_date

 , pers.last\_update\_date

 , grp\_lca\_info.fcc p\_group\_lca\_fcc

----- , dualcomp.prim\_person\_type\_id primary\_pers\_type\_id

 , DECODE

 ( pers.attribute\_category

 , '6', '6'

 , '123', '123'

 , '142', '142'

 , '143', '143'

 , '144', '144'

 , '145', '145'

 , '146', '146'

 , '147', '147'

 , '208', '144'

 , '209', '144'

 , '210', '144'

 , '211', '143'

 , '212', '143'

 , '213', '143'

 , '214', '143'

 , '215', '123'

 , '216', '123'

 , '217', '123'

 , '218', '123'

 , '219', '142'

 , '220', '142'

 , pers.attribute\_category) primary\_pers\_type\_id

 , pers.attribute8 pers\_civilian\_flag

 , asgn.primary\_flag

 --- end adds Aug-2011

 FROM hr.per\_all\_people\_f pers

 , hr.per\_all\_assignments\_f asgn

 , hr.per\_person\_types ptyp

 , hr.per\_person\_type\_usages\_f ptuse

 , hr.per\_qualifications qual

 , hr.hr\_all\_organization\_units asgn\_org

----- , dod\_person\_types\_temp dualcomp --- add Aug-2011

 , ( SELECT org\_lca.organization\_id

 , DECODE( org\_lca.org\_information\_context

 , 'DOD\_AF\_LCA', 'AIR FORCE'

 , 'DOD\_NAVY\_LCA', 'NAVY'

 , 'DOD\_ARMY\_LCA', 'ARMY'

 ) lca\_service

 , org\_lca.start\_date lca\_start\_date

 , TO\_DATE(NVL( org\_lca.end\_date,'31-DEC-4712')) lca\_end\_date

 , org\_lca.dmis\_id

 FROM apps.dod\_lca\_classification\_v org\_lca

 WHERE org\_lca.org\_information\_context

 IN ( 'DOD\_AF\_LCA','DOD\_NAVY\_LCA','DOD\_ARMY\_LCA' )

 ) org\_info

 , ( SELECT pgroup.people\_group\_id

 , pgroup.segment1 pgroup\_organization\_id

 , organztn.organization\_id

 , organztn.attribute13 organization\_uic

 FROM hr.hr\_all\_organization\_units organztn

 , hr.pay\_people\_groups pgroup

 WHERE pgroup.segment1 = organztn.organization\_id

 ) group\_info

 , ( SELECT pgroup.people\_group\_id

 , pgroup.segment1 pgroup\_organization\_id

 , org\_lca.organization\_id

 , DECODE( org\_lca.org\_information\_context

 , 'DOD\_AF\_LCA', 'AIR FORCE'

 , 'DOD\_NAVY\_LCA', 'NAVY'

 , 'DOD\_ARMY\_LCA', 'ARMY'

 ) lca\_service

 , org\_lca.start\_date lca\_start\_date

 , TO\_DATE(NVL( org\_lca.end\_date,'31-DEC-4712')) lca\_end\_date

 , org\_lca.dmis\_id

 , org\_lca.fcc

 FROM hr.pay\_people\_groups pgroup

 , apps.dod\_lca\_classification\_v org\_lca

 WHERE pgroup.segment1 = org\_lca.organization\_id

 ) grp\_lca\_info

 , ( SELECT ppei.person\_id

 , TO\_DATE(ppei.pei\_information1,'YYYY/MM/DD HH24:MI:SS') pers\_eit\_start\_date

 , TO\_DATE(NVL(TO\_DATE(ppei.pei\_information2,'YYYY/MM/DD HH24:MI:SS'),'31-DEC-4712')) pers\_eit\_end\_date

 , jobs.attribute3 dod\_occupation\_code

 , jobs.attribute6 taxonomy\_code

 FROM hr.per\_people\_extra\_info ppei

 , hr.per\_jobs jobs

 WHERE ppei.information\_type = 'DOD\_DUTY\_OCCUPATIONS'

 AND ppei.pei\_information4 = 'PRIMARY'

 AND ppei.pei\_information3 = jobs.job\_id

 ) per\_eit\_info

 WHERE asgn.person\_id = pers.person\_id

-- start mods March 2012 restrict the data to only recent records with mod 1

 AND 10\*trunc(MOD(asgn.person\_id,100)/10) = l\_mod

 AND (asgn.LAST\_UPDATE\_DATE >= vc\_extract\_effect\_date or pers.last\_update\_date >= vc\_extract\_effect\_date)

-- end mods March 2012

 AND asgn.effective\_end\_date

 BETWEEN vc\_extract\_effect\_date AND vc\_hr\_max\_end\_date

 ----- mods Aug-2011:

 ----- exclude records which were modified after Sept. 2011

 AND asgn.effective\_start\_date < vc\_rec\_change\_date

 ---AND asgn.effective\_start\_date < vc\_todays\_date

 ----- end mods Aug-2011

 AND DECODE( asgn.effective\_end\_date

 , vc\_hr\_max\_end\_date, vc\_todays\_date

 , asgn.effective\_end\_date)

 BETWEEN pers.effective\_start\_date AND pers.effective\_end\_date

 AND asgn.assignment\_type = 'E'

 AND asgn.primary\_flag = 'Y'

 AND pers.person\_id = ptuse.person\_id

 AND ptuse.person\_type\_id = ptyp.person\_type\_id

 AND DECODE( asgn.effective\_end\_date

 , vc\_hr\_max\_end\_date, vc\_todays\_date

 , asgn.effective\_end\_date)

 BETWEEN ptuse.effective\_start\_date AND ptuse.effective\_end\_date

 AND pers.person\_id = qual.person\_id (+)

 AND asgn.organization\_id = asgn\_org.organization\_id

 AND asgn.organization\_id = org\_info.organization\_id (+)

 AND DECODE( asgn.effective\_end\_date

 , vc\_hr\_max\_end\_date, vc\_todays\_date

 , asgn.effective\_end\_date)

 BETWEEN org\_info.lca\_start\_date (+) AND org\_info.lca\_end\_date (+)

 AND asgn.people\_group\_id = group\_info.people\_group\_id (+)

 AND asgn.people\_group\_id = grp\_lca\_info.people\_group\_id (+)

 AND DECODE( asgn.effective\_end\_date

 , vc\_hr\_max\_end\_date, vc\_todays\_date

 , asgn.effective\_end\_date)

 BETWEEN grp\_lca\_info.lca\_start\_date (+) AND grp\_lca\_info.lca\_end\_date (+)

 AND asgn.person\_id = per\_eit\_info.person\_id (+)

 AND DECODE( asgn.effective\_end\_date

 , vc\_hr\_max\_end\_date, vc\_todays\_date

 , asgn.effective\_end\_date)

 BETWEEN per\_eit\_info.pers\_eit\_start\_date (+) AND per\_eit\_info.pers\_eit\_end\_date (+)

 ----- mods Aug-2011: need mapping from person-type in pers-dff to single component

----- AND pers.attribute\_category = dualcomp.person\_type\_id

 ----- end mods Aug-2011

 ORDER BY pers.person\_id

 , pers.effective\_start\_date

 , ptuse.effective\_start\_date

 , ptuse.effective\_end\_date

 , ptuse.object\_version\_number DESC

 ;

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 /\*\*\*\*\*\*\*\*\*\*\*\*\* Local Variables \*\*\*\*\*\*\*\*\*\*\*\*/

 v\_output\_record VARCHAR2(1000);

 v\_record\_count NUMBER;

 v\_prev\_person\_id NUMBER;

 v\_prev\_collectn\_person\_id NUMBER;

 v\_prev\_ptype\_usage\_start\_dt per\_person\_type\_usages\_f.effective\_start\_date%TYPE;

 v\_prev\_ptype\_usage\_end\_dt per\_person\_type\_usages\_f.effective\_end\_date%TYPE;

 v\_prev\_ptype\_usage\_ovn per\_person\_type\_usages\_f.object\_version\_number%TYPE;

 v\_organization\_uic hr.hr\_all\_organization\_units.attribute13%TYPE;

 v\_person\_uic hr.per\_all\_people\_f.attribute5%TYPE;

 v\_pers\_group\_organization\_uic hr.hr\_all\_organization\_units.attribute13%TYPE;

 v\_service\_unique\_occ\_code VARCHAR2(50);

 TYPE empl\_info\_rec\_tbl

 IS TABLE OF c\_empl\_info\_recs%ROWTYPE

 INDEX BY PLS\_INTEGER;

 empl\_rec empl\_info\_rec\_tbl;

 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 --------------------------------------------------------------------------------

 --------------------------------------------------------------------------------

 FUNCTION service\_unique\_occ\_code

 ( x\_person\_id IN per\_all\_people\_f.person\_id%TYPE

 , x\_service\_type IN VARCHAR2

 , x\_effective\_date IN DATE

 , x\_person\_type\_id IN per\_all\_people\_f.attribute\_category%TYPE

 , x\_grade IN per\_all\_people\_f.attribute2%TYPE

 , x\_primary\_flag IN per\_all\_assignments\_f.primary\_flag%TYPE

 , x\_skill\_type IN per\_all\_assignments\_f.ass\_attribute5%TYPE

 , x\_skill\_type\_suffix IN per\_all\_assignments\_f.ass\_attribute6%TYPE

 , x\_job\_id IN per\_all\_assignments\_f.job\_id%TYPE

 , x\_civilian\_flag IN per\_all\_people\_f.attribute8%TYPE

 ) RETURN VARCHAR2 IS

 CURSOR c\_skill\_level\_info IS

 SELECT attribute2

 FROM per\_grades

 WHERE name = x\_grade;

 CURSOR c\_assign\_job IS

 SELECT pjd.segment2 job\_occupation\_type

 , pjd.segment3 job\_series

 FROM per\_job\_definitions pjd

 , per\_jobs pj

 WHERE pjd.job\_definition\_id = pj.job\_definition\_id

 AND pj.job\_id = x\_job\_id;

 CURSOR c\_people\_job (c\_occ\_type VARCHAR2) IS

 SELECT pjd.segment2

 , pjd.segment3

 , pei.pei\_information5

 , jdef2.segment3 adt\_qual1

 FROM per\_job\_definitions pjd

 , per\_jobs pj

 , per\_people\_extra\_info pei

 , per\_jobs job2

 , per\_job\_definitions jdef2

 WHERE pjd.job\_definition\_id = pj.job\_definition\_id

 AND pj.job\_id = pei.pei\_information3

 AND TRUNC(x\_effective\_date)

 BETWEEN TO\_DATE(SUBSTR(pei\_information1,1,10),'YYYY/MM/DD')

 AND TO\_DATE(NVL(SUBSTR(pei\_information2,1,10),'4712/12/31'),'YYYY/MM/DD')

 AND person\_id = x\_person\_id

 AND pei\_information4 = c\_occ\_type

 AND pei.information\_type = 'DOD\_DUTY\_OCCUPATIONS'

 AND pei.pei\_information5 = job2.job\_id (+)

 AND job2.job\_definition\_id = jdef2.job\_definition\_id (+)

 ORDER BY pjd.segment2;

 CURSOR c\_people\_job1 IS

 SELECT pjd.segment3

 FROM per\_job\_definitions pjd

 , per\_jobs pj

 , per\_people\_extra\_info pei

 WHERE pjd.segment2 = 'NQ'

 AND pjd.job\_definition\_id = pj.job\_definition\_id

 AND pj.job\_id = pei.pei\_information3

 AND TRUNC(x\_effective\_date)

 BETWEEN TO\_DATE(SUBSTR(pei\_information1,1,10),'YYYY/MM/DD')

 AND TO\_DATE(NVL(SUBSTR(pei\_information2,1,10),'4712/12/31'),'YYYY/MM/DD')

 AND person\_id = x\_person\_id

 AND information\_type = 'DOD\_DUTY\_OCCUPATIONS';

 CURSOR c\_people\_job2 (c\_occ\_type VARCHAR2) IS

 SELECT pjd.segment3

 FROM per\_job\_definitions pjd

 , per\_jobs pj

 , per\_people\_extra\_info pei

 WHERE pjd.segment2 = 'NS'

 AND pjd.job\_definition\_id = pj.job\_definition\_id

 AND pj.job\_id = pei.pei\_information3

 AND TRUNC(x\_effective\_date)

 BETWEEN TO\_DATE(SUBSTR(pei\_information1,1,10),'YYYY/MM/DD')

 AND TO\_DATE(NVL(SUBSTR(pei\_information2,1,10),'4712/12/31'),'YYYY/MM/DD')

 AND person\_id = x\_person\_id

 AND pei\_information4 = c\_occ\_type

 AND information\_type = 'DOD\_DUTY\_OCCUPATIONS';

 l\_result VARCHAR2(50);

 l\_person\_category VARCHAR2(1) := NULL;

 l\_pers\_ns\_occ\_series per\_job\_definitions.segment3%TYPE;

 l\_pers\_occ\_occ\_type per\_job\_definitions.segment2%TYPE;

 l\_pers\_occ\_series per\_job\_definitions.segment3%TYPE;

 l\_pers\_occ\_job\_id per\_people\_extra\_info.pei\_information5%TYPE;

 l\_asgn\_occ\_type per\_job\_definitions.segment2%TYPE;

 l\_asgn\_series per\_job\_definitions.segment3%TYPE;

 l\_skill\_level per\_grades.attribute2%TYPE;

 l\_aqd\_series VARCHAR2(50) DEFAULT NULL;

 l\_adt\_qual1 VARCHAR2(50) DEFAULT NULL;

 l\_occupation\_type VARCHAR2(50) := NULL;

 BEGIN -- service\_unique\_occ\_code

 IF x\_person\_type\_id = '145'

 OR x\_person\_type\_id = '146'

 THEN

 l\_occupation\_type := 'PRIMARY RESERVIST';

 ELSE

 IF x\_primary\_flag = 'Y'

 THEN

 l\_occupation\_type := 'PRIMARY';

 ELSE

 l\_occupation\_type := 'SECONDARY';

 END IF;

 END IF;

 -- obtain person-category without making a call

 l\_person\_category := NULL;

 IF x\_person\_type\_id = '142'

 THEN

 l\_person\_category := 'V';

 ELSIF x\_person\_type\_id = '147'

 THEN

 l\_person\_category := 'L';

 ELSIF x\_person\_type\_id = '123'

 THEN

 l\_person\_category := 'X';

 ELSIF x\_person\_type\_id in ('144', '145', '146')

 THEN

 IF SUBSTR(x\_grade,1,1) IN ('W', 'O')

 THEN

 l\_person\_category := 'O';

 ELSIF SUBSTR(x\_grade,1,1) = 'E'

 THEN

 l\_person\_category := 'E';

 END IF;

 ELSIF x\_person\_type\_id = '143'

 THEN

 IF x\_civilian\_flag = 'Y'

 THEN

 l\_person\_category := 'T';

 ELSE

 l\_person\_category := 'C';

 END IF;

 END IF;

 -- end person-category

 OPEN c\_assign\_job;

 FETCH c\_assign\_job

 INTO l\_asgn\_occ\_type, l\_asgn\_series;

 CLOSE c\_assign\_job;

 -- Non DOD Civilians and Local Nationals

 IF x\_person\_type\_id = '147'

 OR ( x\_person\_type\_id = '143'

 AND x\_civilian\_flag = 'Y')

 THEN

 IF UPPER(x\_service\_type) = 'NAVY'

 THEN

 IF l\_asgn\_occ\_type = 'OC'

 THEN

 l\_result := l\_asgn\_series || 'C';

 ELSE

 l\_result := l\_asgn\_series;

 END IF;

 ELSE

 l\_result := l\_asgn\_series;

 END IF;

 ELSE

 OPEN c\_people\_job (l\_occupation\_type);

 FETCH c\_people\_job

 INTO l\_pers\_occ\_occ\_type, l\_pers\_occ\_series

 , l\_pers\_occ\_job\_id, l\_adt\_qual1;

 CLOSE c\_people\_job;

 ----------------

 -- Air Force

 IF UPPER(x\_service\_type) = 'AIR FORCE'

 THEN

 IF x\_person\_type\_id IN ('123', '142')

 THEN -- CRS06550

 IF (l\_asgn\_series = '0610' AND x\_skill\_type = '2')

 THEN

 l\_result := l\_asgn\_series || x\_skill\_type || x\_skill\_type\_suffix;

 ELSE

 l\_result := l\_asgn\_series;

 END IF;

 ELSIF (x\_person\_type\_id = '143')

 THEN

 IF (l\_pers\_occ\_series = '0610' AND x\_skill\_type = '2') THEN

 l\_result := l\_pers\_occ\_series || x\_skill\_type || x\_skill\_type\_suffix;

 ELSE

 l\_result := l\_pers\_occ\_series;

 END IF;

 ELSE

 l\_result := l\_pers\_occ\_series;

 END IF;

 -- end Air Force

 ----------------

 -- Army

 ELSIF UPPER(x\_service\_type) = 'ARMY'

 THEN

 IF l\_person\_category = 'E'

 THEN

 IF l\_pers\_occ\_job\_id IS NOT NULL

 THEN

 OPEN c\_skill\_level\_info;

 FETCH c\_skill\_level\_info INTO l\_skill\_level;

 CLOSE c\_skill\_level\_info;

 l\_result := l\_pers\_occ\_series || l\_skill\_level || l\_adt\_qual1;

 ELSE

 l\_result := l\_pers\_occ\_series;

 END IF;

 ELSIF l\_person\_category = 'O'

 THEN

 IF l\_pers\_occ\_job\_id IS NOT NULL

 THEN

 l\_result := l\_pers\_occ\_series || l\_adt\_qual1;

 ELSE

 l\_result := l\_pers\_occ\_series;

 END IF;

 ELSIF l\_person\_category = 'C'

 THEN

 IF (l\_pers\_occ\_series = '0610' AND x\_skill\_type = '2')

 THEN

 l\_result := l\_pers\_occ\_series || x\_skill\_type || x\_skill\_type\_suffix;

 ELSE

 l\_result := l\_pers\_occ\_series;

 END IF;

 ELSIF l\_person\_category IN ('V', 'X')

 THEN

 IF (l\_asgn\_series = '0610' AND x\_skill\_type = '2')

 THEN

 l\_result := l\_asgn\_series || x\_skill\_type || x\_skill\_type\_suffix;

 ELSE

 l\_result := l\_asgn\_series;

 END IF;

 END IF;

 -- end Army

 ----------------

 -- Navy

 ELSIF UPPER(x\_service\_type) = 'NAVY'

 THEN

 IF x\_person\_type\_id IN ('123', '142')

 THEN -- CRS06550

 IF l\_asgn\_series = '0610'

 THEN

 l\_result := l\_asgn\_series || 'C' || x\_skill\_type || x\_skill\_type\_suffix;

 ELSIF l\_asgn\_occ\_type = 'OC'

 THEN

 l\_result := l\_asgn\_series || 'C';

 ELSE

 l\_result := '0000C';

 END IF;

 ELSIF x\_person\_type\_id = '147'

 THEN

 IF l\_pers\_occ\_occ\_type = 'OC'

 THEN

 l\_result := l\_pers\_occ\_series || 'C';

 ELSE

 l\_result := '0000C';

 END IF;

 ELSIF x\_person\_type\_id = '143'

 THEN

 IF l\_pers\_occ\_series = '0610'

 THEN

 l\_result := l\_pers\_occ\_series || 'C' || x\_skill\_type || x\_skill\_type\_suffix;

 ELSIF l\_pers\_occ\_occ\_type = 'OC'

 THEN

 l\_result := l\_pers\_occ\_series || 'C';

 ELSIF l\_pers\_occ\_series IS NOT NULL

 THEN

 l\_result := l\_pers\_occ\_series || 'C';

 ELSE

 l\_result := '0000C';

 END IF;

 ELSE

 IF l\_person\_category = 'O'

 THEN

 OPEN c\_people\_job2 (l\_occupation\_type);

 FETCH c\_people\_job2

 INTO l\_pers\_ns\_occ\_series;

 CLOSE c\_people\_job2;

 IF SUBSTR(l\_pers\_ns\_occ\_series,4,1) = '1'

 AND SUBSTR(l\_pers\_ns\_occ\_series,1,2) IN ('15', '16')

 THEN

 OPEN c\_people\_job1;

 FETCH c\_people\_job1

 INTO l\_aqd\_series;

 CLOSE c\_people\_job1;

 l\_result := SUBSTR(l\_pers\_ns\_occ\_series,1,4)||l\_aqd\_series;

 ELSIF SUBSTR(x\_grade,1,1) = 'O'

 OR x\_grade = 'WO'

 THEN

 l\_result := SUBSTR(l\_pers\_ns\_occ\_series,1,4)||'O';

 END IF;

 IF NVL(l\_result, 'O') = 'O'

 THEN

 l\_result := '0000O';

 END IF;

 ELSIF l\_person\_category = 'E'

 THEN

 IF l\_pers\_occ\_occ\_type = 'NR'

 THEN

 IF l\_pers\_occ\_series IN ('DT')

 THEN

 l\_result := '00'||l\_pers\_occ\_series||'E';

 ELSIF l\_pers\_occ\_series IN ('HA', 'HR', 'HN', 'HM')

 THEN

 l\_result := '00' || 'GDE'; --changed from 'HME' to 'GDE'

 ELSIF l\_pers\_occ\_series IN ('DA', 'DR', 'DN')

 THEN

 l\_result := '00' || 'DTE';

 ELSE

 l\_result := '0000E';

 END IF;

 ELSIF l\_pers\_occ\_occ\_type = 'NE'

 THEN

 l\_result := l\_pers\_occ\_series||'E';

 END IF;

 IF NVL(l\_result, 'E') = 'E'

 THEN

 l\_result := '0000E';

 END IF;

 END IF;

 END IF;

 -- end Navy

 ----------------

 -- Marine Corps

 ELSIF UPPER(x\_service\_type) = 'MARINE CORPS'

 THEN

 IF l\_person\_category = 'O'

 THEN

 IF SUBSTR(x\_grade,1,1) = 'O'

 OR x\_grade = 'WO'

 THEN

 OPEN c\_people\_job2 (l\_occupation\_type);

 FETCH c\_people\_job2

 INTO l\_result;

 CLOSE c\_people\_job2;

 END IF;

 IF l\_result IS NULL

 THEN

 l\_result := '0000O';

 END IF;

 ELSIF l\_person\_category = 'E'

 THEN

 IF SUBSTR(x\_grade,1,1) = 'E'

 THEN

 l\_result := l\_pers\_occ\_series||'E';

 END IF;

 IF l\_result IS NULL

 THEN

 l\_result := '0000E';

 END IF;

 END IF;

 END IF; -- end Marine

 END IF;

 RETURN(l\_result);

 END service\_unique\_occ\_code;

 --------------------------------------------------------------------------------

 --------------------------------------------------------------------------------

BEGIN -- extract main block

 -- write column header

 -- mod Aug. 10, 2011 added 5 columns for taxonomy codes

 DBMS\_OUTPUT.PUT\_LINE

 ('SSN!EDIPN!NPI!Person ID!Emloyee Number!Employee Assignment DMIS ID'

 ||'!Employee People Group DMIS ID!Position ID!Job ID!Organization ID'

 ||'!People Group ID!Assignment Effective Start Date'

 ||'!Assignment Effective End Date!Person Effective Start Date'

 ||'!Person Effective End Date!Original Date of Hire'

 ||'!Person Assignment Service!People Group Service!Last Name!First Name'

 ||'!Middle Name!Suffix!Title Rank!Person Grade!Person Step!Gender!Person Type'

 ||'!Skill Type!Skill Type Suffix!Organization UIC or PAS!People Group UIC'

 ||'!DOD Occupation Code'

 ||'!Taxonomy Code 1!Taxonomy Code 2!Taxonomy Code 3'

 ||'!Taxonomy Code 4!Taxonomy Code 5!Taxonomy Code 6'

 ||'!Employee Service!Employee UIC/PAS'

 ||'!SUOC!Record ID!Creation Date!Last Update Date!People Group FCC'

 );

 v\_record\_count := 0;

 v\_prev\_person\_id := 0;

 v\_prev\_collectn\_person\_id := 0;

 v\_prev\_ptype\_usage\_start\_dt := '01-JAN-1951';

 v\_prev\_ptype\_usage\_end\_dt := vc\_hr\_max\_end\_date;

 v\_prev\_ptype\_usage\_ovn := 0;

 IF c\_empl\_info\_recs%ISOPEN

 THEN

 CLOSE c\_empl\_info\_recs;

 END IF;

 OPEN c\_empl\_info\_recs;

 LOOP -- process cursor

 BEGIN

 -- get collection of records

 FETCH c\_empl\_info\_recs BULK COLLECT INTO empl\_rec LIMIT vc\_bulk\_collect\_limit;

 EXIT WHEN empl\_rec.COUNT = 0;

 -- process data collected

 FOR indx IN 1 .. empl\_rec.COUNT LOOP

 BEGIN

 -- do not process new ex-employees

 -- do not process the person-type-usage record for retiree/beneficiary

 -- get first person-usage-type record ordered by object-version-number

 IF ( empl\_rec(indx).system\_person\_type != 'EX\_EMP'

 OR v\_prev\_person\_id = empl\_rec(indx).person\_id

 )

 AND

 ( empl\_rec(indx).person\_id != v\_prev\_collectn\_person\_id

 OR empl\_rec(indx).pers\_type\_usage\_start\_dt != v\_prev\_ptype\_usage\_start\_dt

 OR empl\_rec(indx).pers\_type\_usage\_end\_dt != v\_prev\_ptype\_usage\_end\_dt

 OR empl\_rec(indx).pers\_type\_usage\_ovn = v\_prev\_ptype\_usage\_ovn

 )

 THEN

 v\_prev\_person\_id := empl\_rec(indx).person\_id;

 v\_record\_count := v\_record\_count + 1;

 -- prefix Navy UICs with an N

 -- initialize without navy denotation

 v\_organization\_uic := empl\_rec(indx).organization\_uic;

 v\_pers\_group\_organization\_uic := empl\_rec(indx).p\_group\_organization\_uic;

 v\_person\_uic := empl\_rec(indx).person\_uic;

 -- only prefix uic when has a value

 IF empl\_rec(indx).organization\_lca\_service = 'NAVY'

 AND empl\_rec(indx).organization\_uic IS NOT NULL

 THEN

 v\_organization\_uic := 'N'|| empl\_rec(indx).organization\_uic;

 END IF;

 IF empl\_rec(indx).p\_group\_lca\_service = 'NAVY'

 AND empl\_rec(indx).p\_group\_organization\_uic IS NOT NULL

 THEN

 v\_pers\_group\_organization\_uic := 'N'|| empl\_rec(indx).p\_group\_organization\_uic;

 END IF;

 IF empl\_rec(indx).person\_service IN ('NAVY','MARINE CORPS')

 AND empl\_rec(indx).person\_uic IS NOT NULL

 THEN

 v\_person\_uic := 'N'|| empl\_rec(indx).person\_uic;

 END IF;

 --- add Aug-2011

 --- function to get the suoc value

 --- code copied and tuned from eas package (dod\_easiv\_ext.return\_suoc)

 v\_service\_unique\_occ\_code

 := service\_unique\_occ\_code

 ( x\_person\_id => empl\_rec(indx).person\_id

 , x\_service\_type => empl\_rec(indx).person\_service

 , x\_effective\_date => empl\_rec(indx).empl\_assign\_start\_date

 , x\_person\_type\_id => empl\_rec(indx).primary\_pers\_type\_id

 , x\_grade => empl\_rec(indx).grade

 , x\_primary\_flag => empl\_rec(indx).primary\_flag

 , x\_skill\_type => empl\_rec(indx).skill\_type

 , x\_skill\_type\_suffix => empl\_rec(indx).skill\_type\_suffix

 , x\_job\_id => empl\_rec(indx).job\_id

 , x\_civilian\_flag => empl\_rec(indx).pers\_civilian\_flag

 );

 --- end add Aug-2011

 -- do have people-group, then get fcc, dmis-id, parent-dmis-id and service

 v\_output\_record :=

 ( empl\_rec(indx).national\_identifier

 || vc\_ascii\_delimiter || empl\_rec(indx).edipn

 || vc\_ascii\_delimiter || empl\_rec(indx).npi

 || vc\_ascii\_delimiter || empl\_rec(indx).person\_id

 || vc\_ascii\_delimiter || empl\_rec(indx).employee\_number

 || vc\_ascii\_delimiter || empl\_rec(indx).organization\_lca\_dmis\_id

 || vc\_ascii\_delimiter || empl\_rec(indx).p\_group\_lca\_dmis\_id

 || vc\_ascii\_delimiter || empl\_rec(indx).position\_id

 || vc\_ascii\_delimiter || empl\_rec(indx).job\_id

 || vc\_ascii\_delimiter || empl\_rec(indx).organization\_id

 || vc\_ascii\_delimiter || empl\_rec(indx).people\_group\_id

 || vc\_ascii\_delimiter || TO\_CHAR(empl\_rec(indx).empl\_assign\_start\_date,'DD-MON-YYYY')

 || vc\_ascii\_delimiter || TO\_CHAR(empl\_rec(indx).empl\_assign\_end\_date,'DD-MON-YYYY')

 || vc\_ascii\_delimiter || TO\_CHAR(empl\_rec(indx).empl\_hr\_start\_date,'DD-MON-YYYY')

 || vc\_ascii\_delimiter || TO\_CHAR(empl\_rec(indx).empl\_hr\_end\_date,'DD-MON-YYYY')

 || vc\_ascii\_delimiter || TO\_CHAR(empl\_rec(indx).orig\_date\_of\_hire,'DD-MON-YYYY')

 || vc\_ascii\_delimiter || empl\_rec(indx).organization\_lca\_service

 || vc\_ascii\_delimiter || empl\_rec(indx).p\_group\_lca\_service

 || vc\_ascii\_delimiter || empl\_rec(indx).last\_name

 || vc\_ascii\_delimiter || empl\_rec(indx).first\_name

 || vc\_ascii\_delimiter || empl\_rec(indx).middle\_names

 || vc\_ascii\_delimiter || empl\_rec(indx).suffix

 || vc\_ascii\_delimiter || empl\_rec(indx).title\_rank

 || vc\_ascii\_delimiter || empl\_rec(indx).grade

 || vc\_ascii\_delimiter || empl\_rec(indx).step

 || vc\_ascii\_delimiter || empl\_rec(indx).gender

 || vc\_ascii\_delimiter || empl\_rec(indx).person\_type

 || vc\_ascii\_delimiter || empl\_rec(indx).skill\_type

 || vc\_ascii\_delimiter || empl\_rec(indx).skill\_type\_suffix

 || vc\_ascii\_delimiter || v\_organization\_uic

 || vc\_ascii\_delimiter || v\_pers\_group\_organization\_uic

 || vc\_ascii\_delimiter || empl\_rec(indx).dod\_occupation\_code

 || vc\_ascii\_delimiter || empl\_rec(indx).taxonomy\_code

 -- mod Aug. 10, 2011 added 5 columns for taxonomy codes

 || vc\_ascii\_delimiter || NULL -- taxonomy 2

 || vc\_ascii\_delimiter || NULL -- taxonomy 3

 || vc\_ascii\_delimiter || NULL -- taxonomy 4

 || vc\_ascii\_delimiter || NULL -- taxonomy 5

 || vc\_ascii\_delimiter || NULL -- taxonomy 6

 || vc\_ascii\_delimiter || empl\_rec(indx).person\_service

 || vc\_ascii\_delimiter || v\_person\_uic

 || vc\_ascii\_delimiter || v\_service\_unique\_occ\_code

 || vc\_ascii\_delimiter || empl\_rec(indx).person\_rowid

 || vc\_ascii\_delimiter || TO\_CHAR(empl\_rec(indx).creation\_date,'DD-MON-YYYY')

 || vc\_ascii\_delimiter || TO\_CHAR(empl\_rec(indx).last\_update\_date,'DD-MON-YYYY')

 || vc\_ascii\_delimiter || empl\_rec(indx).p\_group\_lca\_fcc

 );

 DBMS\_OUTPUT.PUT\_LINE( v\_output\_record );

 END IF; -- test for new ex-employee

 v\_prev\_collectn\_person\_id := empl\_rec(indx).person\_id;

 v\_prev\_ptype\_usage\_start\_dt := empl\_rec(indx).pers\_type\_usage\_start\_dt;

 v\_prev\_ptype\_usage\_end\_dt := empl\_rec(indx).pers\_type\_usage\_end\_dt;

 v\_prev\_ptype\_usage\_ovn := empl\_rec(indx).pers\_type\_usage\_ovn;

 END;

 END LOOP; -- collection loop

 END;

 END LOOP; -- process cursor

 CLOSE c\_empl\_info\_recs;

 DBMS\_OUTPUT.PUT\_LINE('This record must be here - Complete: ' || v\_record\_count);

COMMIT;

END;

/

spool off