**5 March 2019**

CDR Medications

for the

MHS Data Repository (MDR)

(Version 1.02.00)

Current Specification

Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Originator | Para/Tbl/Fig | Description of Change |
| 1.00.00 | 01/17/2011 | C. Kangas |  | Baseline |
| 1.01.00 | 11/19/2018 | N. Bowling | Table 5 | Adjusted for NDAA-related changes in the LVM. |
| 1.02.00 | 3/5/2019 | N. Bowling | Table 5 | Adjusted the logic for the ACV Group to be set to blank for dates on or after 1/1/2019. |

# MDR Medication Tables

1. Background

This specification describes the process required to create the MDR Medication tables based on data received from the Clinical Data Repository (CDR). The scope of the medication data contained in the CDR is medication orders and fills that occur at MTFs. The MDR has contained PDTS data files for many years, generated from extracts received from the PDTS data warehouse. The addition of this CDR-based medication information is not meant to replace the PDTS data already available in the MDR, but to compliment it. One major reason for placing the CDR Medication data in the MDR is that the PDTS data does not contain information related to inpatient medications, whereas the CDR does contain medication ordering and fill information for inpatient care.

1. Sources

The source data files used to create the MDR Medication tables are extracted from the AHLTA CDR. The transfer of the raw source extracts is handled by DHSS for loading into the MDR for further processing. These files are listed below:

**Table 1. CDR Sources**

| **CDR Source** | **Data Files** | **Purpose** |
| --- | --- | --- |
| CDR Medication Orders Table | RXMEO\*.DAT | Collection of records for direct care Medication Orders in raw text form, prepared in accordance with the ICD. |
| CDR Medication Fills Table | RXMEF\*.DAT | Collection of records for direct care Medication Fills in raw text form, prepared in accordance with the ICD. |

1. Transmission (Format and Frequency)

Source files are provided according to the frequency described in the Table 2. The format of these feeds is described in ICD XXXX.

**Table 2. Frequency of CDR Source Files**

|  |  |
| --- | --- |
| **Source File** | **Frequency** |
| CDR Medication Orders Table | Weekly |
| CDR Medication Fills Table | Weekly |

1. Organization and batching

Source Data: The first step in MDR processing is to batch records received from CDR. Raw data batches are stored in MDR\RAW according to routine MDR operating procedures.

Output Products: There are two final MDR Medication files that are output by the processor:

* Medication Fills: The Fills table is a single FY level SAS dataset. Each record represents a single fill of a medication order. The processor needs to be run once for each FY to be processed. The processor performs minimal field derivations and no external file merges to create this file. This file is intended to be used in combination with the Orders table.
* Medication Orders: The Orders table is also a single FY level SAS dataset. Each record represents a unique medication order made at an MTF. Fill information is merged into the final MDR Medication Orders table, so the Fills processor needs to be run first before the Orders processor is run each week. The processor needs to be run once for each FY to be processed. The processor performs several external merges and many field derivations, and must also apply updates to Order records across extracts.

The current fiscal year is processed weekly, and past fiscal years are processed on a less frequent basis (See Section VIII for refresh frequency). Table 3 contains the location and names of the output products. The preparation of them is described in subsequent sections of this document.

**Table 3: MDR Medication Processor Output Products**

|  |  |  |
| --- | --- | --- |
| **Medication Processor Output** | **File Naming Convention** | **Member Name** |
| MDR Medication Orders File  | /mdr/pub/cdr/med/fy<yy> | orders.sas7bdat |
| MDR Medication Fills File  | /mdr/pub/cdr/med/fy<yy> | fills.sas7bdat |

Archival of files is also required, so that corresponding “apub” and other processing files (i.e. log, aprod, etc) are also loaded into the MDR according to routine operating procedures.

1. Receiving Filters
2. The feed files sent to the MDR are tilde (~) delimited files. In the feeds, there are free text fields that occasionally contain tildes within them, which disrupts the normal ingest function and structure of fields in a delimited file. In these cases, the processor reads in the field created by the inadvertent extra delimiter and drops it. For the "good" field that had the extra tilde within it, this method only keeps the content of the field up to the tilde embedded in it; any content after the tilde in that field is dropped.
3. Only raw records with order or fill dates in FY09 or later are kept.
4. Field Transformations and Deletions for MDR
5. During the extraction of the raw orders records, de-duplication of records, or anytime a medication order key collision occurs between incoming data and existing master data, the processor de-duplicates data by selecting the largest value of the Feed Date (FEEDDATE) for any given order key (HOSTDMIS + MED\_ORDER\_ID). If multiple records exist with the same FEEDDATE and order key, the record with the largest ROW\_NUMBER is kept.

The final record that remains for a given order is then kept or deleted based on the ACTION field. For records with an ACTION = “D” (delete), the record is removed from the MDR Orders dataset. For all other records with an ACTION = “I” (insert) or ACTION = “U” (update), the records are kept in the MDR Orders dataset.

1. During the extraction of the raw fills records, de-duplication of records, or anytime a medication fill key collision occurs between incoming data and existing master data, the processor de-duplicates data by selecting the largest value of the Feed Date (FEEDDATE) for any given fill key (HOSTDMIS + MED\_ORDER\_ID + REM\_REFILLS). If multiple records exist with the same FEEDDATE and fill key, the record with the latest DATE\_DISPENSED is kept. If multiple records still exist with the same FEEDDATE, fill key, and dispense date, the record with the largest ROW\_NUMBER is kept.
2. There are several merges and formats required to prepare the MDR Medication tables as described in Table 4.

**Table 4: External File Merges and Formats**

| **Merge** | **Merge to** | **Date Matching** | **Additional Matching** |
| --- | --- | --- | --- |
| CDR Patient Table | Orders Table | Most recent CDR Patient Table is used for all FY Medication files. | See CDR Patient Specification. |
| Longitudinal VM File | Orders Table | Date Start, with begin and end dates for each changeable demographic segment | EDI\_PN if available. |
| CHCS Host Format | Orders and Fills Tables | None | Apply the format to host\_facility\_id, which will return HOSTDMIS. |
| DMIS ID Format | Orders Table | None | Apply the format to cdr\_clinic\_id, which will return the DMISID. |
| CDR Medication Fill Table | Orders Table | None | Merge based on HOSTDMIS and MED\_ORDER\_ID. All fills that occur in a different FY are captured.  |

1. Business rules for each of the derived and appended fields that result from file merges and formats are described in the body of Tables 5 and 6.
2. record layout and content

The MDR Medication Order tables are stored as one SAS dataset per fiscal year. The dataset is prepared according to the derivation rules listed in Table 5.

**Table 5. MDR Medication Order SAS Data Set**

| **Variable Name** | **SAS Name** | **Format** | **Transformation Rule** |
| --- | --- | --- | --- |
| CHCS Host | HOSTDMIS | $4. | Derived from application of the CDR Host DMISID format:hostdmis = put(host\_facility\_id,hostdmis.);See Appendix A for the hostdmis format. |
| Medication Order ID | MED\_ORDER\_ID | $12. | Derived using first 12 characters of MED\_ORDER\_ID. |
| Date Start | DATE\_START | $8. | Derived using first 8 characters of DATE\_START.  |
| Date End | DATE\_END | $8. | Derived using first 8 characters of DATE\_END. |
| Fiscal Year | FY | $4. | Fiscal year equivalent of calendar year of Date Start. |
| Fiscal Month | FM | $2. | Fiscal month equivalent of calendar month of Date Start. |
| Treatment DMISID | DMISID | $4. | Obtained from application of the CDR DMISID format:DMISID = put(cdr\_clinic\_id,cdr\_dmis.);  |
| Order Date | MED\_ORDER\_DATE | $8. | Derived using first 8 characters of MED\_ORDER\_DATE. Unavailable before 12/09 |
| Order Status | ORDER\_STATUS | $2. | No transformation. |
| Order Type | ORDER\_TYPE | $2. | Derived as: if order\_type\_id = . then order\_type = “00”; else order\_type = put(order\_type\_id, z2.); |
| Order IEN | ORDER\_IEN | 8. | No transformation. Unavailable < 12/09 |
| Priority | PRIORITY | $1. | Use 1 char of PRIORITY from the feed:A = ASAPP = PREOPR = ROUTINES = STATblank = UNREPORTED |
| Medication NCID | MEDICATION\_NCID | 8. | No transformation. |
| National Drug Classification | NDC | $11. | No transformation. |
| Medication Form | MED\_FORM | $3. | Prior to 2009-12-04 feed, only MED\_FORM\_TEXT was available. After 2009-12-04, no transformation using GIVE\_FORM\_CODE. Derivation of med\_form\_text prior to 2009-12-04 used a medication form format with the logic:if feeddt lt "20091204" then med\_form = put(med\_form\_text, $med\_form.); |
| Medication Route | MED\_ROUTE | $4. | Derived using first 4 characters of MED\_ROUTE\_TEXT. |
| Original Number of Refills on Script | REFILLS | 3. | No transformation. |
| Dispense Amount | DISPENSE\_AMOUNT | 8. | No transformation |
| Give Amount | GIVE\_AMOUNT | 8. | No transformation. Unavailable < 12/09 |
| Duration | DURATION | 8. | No transformation |
| Child Resistant | CHILD\_RESISTANT | $1. | No transformation |
| Group ID | GROUP\_ID | 8. | No transformation |
| Component | COMPONENT | $1. | No transformation |
| Interval Repeat | INTERVAL\_REPEAT | $10. | No transformation |
| SIG Code | SIG\_CODE | $100. | Derived using first 100 characters of SIG\_CODE from feed. Unavailable before 12/09. |
| Dispense Location | DISPENSE\_LOCATION | $30. | No transformation. |
| Pharmacy NPI | PHARMACY\_NPI | $10. | No transformation. |
| Units | UNITS | $12. | Derived using first 12 characters from UNITS\_TEXT. Unavailable before 12/09. |
| Feed Date | FEEDDT | $8. | Derived from the file name of the feed.if substr(file\_info,length(file\_info)-2,3) = ".gz" then feeddt = substr(file\_info,length(file\_info)-23,8);else if substr(file\_info,length(file\_info)-2,3) = "DAT" then feeddt = substr(file\_info,length(file\_info)-20,8); |
| CDR Patient ID | CDR\_PATIENT\_ID | $20. | No transformation (UNIT\_NUMBER) |
| CDR Appointment ID | CDR\_APPT\_ID | $20. | No transformation (APPT\_ID) |
| CDR Clinic ID | CDR\_CLINIC\_ID | $20. | No transformation (CLINIC\_ID) |
| CDR Provider ID | CDR\_PROVIDER\_ID | $20. | No transformation (PROVIDER\_ID) |
| **Medication Fills Merge** |
| Number of Fills | NUM\_FILLS | 3. | No transformation. |
| Remaining Refills | REM\_REFILLS | 3. | No transformation. |
| Last Dispense Date | LAST\_DISPENSE\_DT | $8. | No transformation. |
| **CDR Patient Table Merge** |
| Universal Patient ID | UPID | $14. | Derived from the Patient table merge based on CDR\_PATIENT\_ID |
| EDIPN | EDI\_PN | $10. | Derived from the Patient table merge based on CDR\_PATIENT\_ID |
| Patient SSN | PATSSN | $9. | Derived from the Patient table merge based on CDR\_PATIENT\_ID |
| Sponsor SSN | SPONSSN | $9. | Derived from the Patient table merge based on CDR\_PATIENT\_ID |
| Patient Date of Birth | PATDOB | 8. | Derived from the Patient table merge based on CDR\_PATIENT\_ID |
| Patient Age | PATAGE | 8. | Derived using aprod/util macro by subtracting PATDOB from Date Start |
| Patient Category | PATCAT | $3. | Derived from the Patient table merge based on CDR\_PATIENT\_ID |
| Age Group Code | AGEGRP | $1. | Derived using aprod/util macro. A = 0-17, B = 18-24, etc. |
| **LVM Table Merge** |
| DEERS Gender | GENDER | $1. | Fill with gender from LVM based on EDIPN.If the gender is blank or U, set to “Z”. |
| DEERS Enrollment DMISID | DENRSITE | $4. | Fill with enrollment DMISID from LVM based on EDIPN, if the order start date is between the begin and end date associated with the enrollment site. If no match for the person, set to blank.See VM6 Specification, Exhibits G-18 and 19 for segment and field positions. |
| DEERS Beneficiary Category | BENCAT | $3. | Fill with DEERS beneficiary category from LVM based on EDIPN, if the order start date is between the begin and end date associated with the DEERS beneficiary category. If no match for the person or the bencat is Z, set to “UNK”.See VM6 Specification, Exhibits G-18 and 19 for segment and field positions. |
| DEERS Common Beneficiary Category | COMBEN | $1. | Derived from DEERS Beneficiary Category during LVM merge using MDR utility programs. See VM6 Specification, section A.1.12 for derivation. If no match for the person, set to “3”. |
| DEERS Sponsor Service | DSPONSVC | $1. | Fill with DEERS sponsor service from LVM based on EDIPN, if the order start date is between the begin and end date associated with the DEERS sponsor service. If no match for the person, set to blank.See VM6 Specification, Exhibits G-18 and 19 for segment and field positions. |
| DEERS Sponsor Service Aggregate | DSVCAGG | $1. | Fill with DEERS sponsor service (aggregate) from LVM based on EDIPN, if the order start date is between the begin and end date associated with the DEERS sponsor service (aggregate). If no match for the person, set to blank.See VM6 Specification, Exhibits G-18 and 19 for segment and field positions. |
| DEERS Alternate Care Value | ACV | $1. | Fill with ACV from LVM based on EDIPN, if the order start date is between the begin and end date associated with the ACV, else if ACV is blank after LVM merge and bencat is ACT or GRD then set ACV to M, otherwise set to blank. This field is no longer populated starting 1/1/18.See VM6 Specification, Exhibits G-18 and 19 for segment and field positions. |
| DEERS Relationship | RELATIONSHIP | $1. | Fill with DEERS Relationship from the LVM based on EDIPN and SPONSSN. If Relationship not found in LVM merge, then derive from FMP from Patient Table Merge.See VM6 Specification, Exhibits G-18 and 19 for segment and field positions. |
| DEERS Enrollment HCDP | HCDP | $3. | Fill with DEERS HCDP code from LVM based on EDIPN, if the order start date is between the begin and end date associated with the DEERS HCDP code. If no match for the person, set to blank.See VM6 Specification, Exhibits G-18 and 19 for segment and field positions. |
| DEERS ZIP Code | DEERSZIP | $5. | Fill with DEERS ZIP code from LVM based on EDIPN, if the order start date Is between the begin and end date associated with the DEERS ZIP code. If no match for the person, set to blank.See VM6 Specification, Exhibits G-18 and 19 for segment and field positions. |
| DEERS Eligibility Group | ELG\_GRP | $2. | Fill with Eligibility Group from LVM if the order start date is between the begin and end date of the associated segment. If no match for person, set to “Z”.See VM6 Beneficiary Specification, Exhibit G19. |
| DEERS Enrollment Group | ENR\_GRP | $2. | Fill with Enrollment Group from LVM if the order start date is between the begin and end date of the associated segment. If no match for the person, set to “Z”. See VM6 Beneficiary Specification, Exhibit G19. |
| DEERS PCM Type | PCM\_TYPE | $1. | Fill with PCM Type from LVM if the order start date is between the begin and end date of the associated segment. If no match for the person, set to “Z”. See VM6 Beneficiary Specification, Exhibit G19. |
| DEERS Assigned Health Care Delivery Program Code | HCDP\_ASSGN | $3. | Fill with Assigned HCDP Code from LVM if the order start date is between the begin and end date of the associated segment. See VM6 Beneficiary Specification, Exhibit G19. |
| DEERS ACV Group | ACVGROUP | $2. | For dates on or after 1/1/2019:Set to blank.For dates prior to 1/1/2019:Derived by the MDR utilities during LVM merge based on Enrollment Group, PCM Type, Eligibility Group, and Common Beneficiary Category or ACV and Common Beneficiary Category depending on whether order start date is before or after 1/1/18. If no match for the person, set to “O”. See VM6 Beneficiary Specification, Section G.3 for details. |

The Table 6 contains the file layout for the MDR Medication Fills dataset.

**Table 6. MDR Medication Fills SAS Data Set**

| **Variable Name** | **SAS Name** | **Format** | **Transformation Rule** |
| --- | --- | --- | --- |
| CHCS Host | HOSTDMIS | $4. | Derived from application of the CDR Host DMISID format: hostdmis = put(host\_facility\_id,hostdmis.);See Appendix A for more detail. |
| Medication Order ID | MED\_ORDER\_ID | $12. | Derived using first 12 characters of MED\_ORDER\_ID. |
| Dispense Date | DISPENSE\_DT | $8. | Derived using first 8 characters of DATE\_DISPENSED.  |
| Remaining Refills | REM\_REFILLS | 8. | No transformation. |
| RX Number | RX\_NUMBER | $10. | Derived using first 10 characters of RX\_NUMBER. |
| Fiscal Year | FY | $2. | Fiscal year equivalent of calendar year of Dispense Date. |
| Fiscal Month | FM | $4. | Fiscal month equivalent of calendar month of Dispense Date. |
| Feed Date | FEEDDT | $8. | Derived from the file name of the feed.if substr(file\_info,length(file\_info)-2,3) = ".gz" then feeddt = substr(file\_info,length(file\_info)-23,8);else if substr(file\_info,length(file\_info)-2,3) = "DAT" then feeddt = substr(file\_info,length(file\_info)-20,8); |

1. Refresh Frequency

Frequency of updates (based on Medication Order Start Date):

* Weekly for current FY.
* For the previous FY, weekly for 1 quarter (October, November, and

December), then switch to semiannually (April, October).

* All years prior to prior FY: Semiannually (April, October)
* Retrofits: On an as needed basis when data corrections or updates are

required.

1. Data Quality

It is expected that when the Medication processor is run each week, that basic quality checks are performed throughout the process. It is recommended that the DHSS vendor develop a spreadsheet which tracks key characteristics of the data across processing cycles; making it relatively easy to understand how the data should generally look. DHSS vendors need to review these statistics each month prior to releasing the data. DHCAPE (the functional proponent and the specification author) should be contacted immediately should any quality issues arise. These checks, at a minimum, should include:

* Total record counts in the data feed should have a relatively stable distribution across Medication Order Start Date, accounting for weekends and holidays. Any anomalies should immediately be investigated.
* The number of records ‘cleaned out’ each month should be similar in scope and proportion across update cycles.
* The number of records that match when doing the CDR Patient table merge should be consistent.
* The distribution of all categorical fields (ex. DMISID, ORDER\_STATUS) should be consistent. The results of proc freq analyses will verify this.
* The number of null values for important fields such as CDR\_PATIENT\_ID, MED\_ORDER\_ID, and MEDICATION\_NCID should be tracked across monthly updates.
* When reading in the Medication data feeds, a small number of records should be printed off and manually inspected to ensure they have read in properly.
* Cross tabulations should be reviewed on derived elements to ensure the derivation logic works.
* A data flow tracker should be built to ensure that all records that are intended to make it into the final Medication datasets do. In other words, all inserts, updates, and deletions should be tracked and explained in the data flow worksheet.

**Appendix A: Description of HOSTDMIS format**

The raw CDR feeds contain a CDR unique ID (HOST\_FACILITY\_ID) for the CHCS host, which is not common to any other data table within the MDR. Therefore a SAS format was created to translate the CDR host facility ID to a DMIS ID, a field commonly used in the MDR.

The application of the hostdmis format to translate the host\_facility\_id into hostdmis is done with the following statement: hostdmis = put(host\_facility\_id,hostdmis.);

Below is the proc format code that is used to develop the hostdmis SAS format:

**proc** **format**;

value hostdmis

**76313**='0364'

**76318**='0109'

**76323**='0128'

**76328**='0118'

**76333**='0110'

**76338**='0062'

**76810**='1170'

**1046961**='0052'

**1048021**='0090'

**1049621**='0124'

**1059821**='0089'

**1067401**='0125'

**1074201**='0091'

**1097342**='0248'

**1097429**='0018'

**1097561**='0013'

**1097861**='0055'

**1098981**='0338'

**1099041**='0114'

**1099139**='0098'

**1099332**='0096'

**1099822**='0113'

**1100881**='0108'

**1101099**='0029'

**1104242**='0097'

**1104381**='0112'

**1104541**='0014'

**1105841**='0064'

**1106441**='0028'

**1106901**='0131'

**1107161**='0019'

**1107201**='0024'

**1112813**='0057'

**1113124**='0049'

**1113704**='0048'

**1120878**='0047'

**1130428**='0060'

**1132134**='0038'

**1132684**='0039'

**1134172**='0032'

**1135465**='0103'

**1137626**='0101'

**1138685**='0053'

**1138927**='0056'

**1143097**='0073'

**1144654**='0009'

**1145022**='0045'

**1145350**='0067'

**1177297**='0008'

**1178200**='0001'

**1178583**='0330'

**1180847**='0003'

**1181105**='0058'

**1181588**='0607'

**1185029**='0061'

**1187857**='0075'

**1195255**='0035'

**1200322**='0005'

**1208940**='0612'

**1209517**='0086'

**1214474**='0615'

**1214671**='0010'

**1214914**='0129'

**1215101**='0616'

**1215502**='0620'

**1216727**='0621'

**1217255**='0624'

**1217474**='0618'

**1217695**='0084'

**1217869**='0077'

**1217983**='0085'

**1218117**='0006'

**1218586**='0050'

**1218870**='0119'

**1219060**='0617'

**1219293**='0059'

**1219472**='0808'

**1219659**='0310'

**1224255**='0079'

**1224847**='0093'

**1224981**='0004'

**1225163**='0076'

**1225324**='0078'

**1225841**='0106'

**1226061**='0635'

**1226215**='0637'

**1226261**='0633'

**1226659**='0083'

**1226824**='0623'

**1226983**='0638'

**1227261**='0074'

**1227781**='0094'

**1228014**='0042'

**1228561**='0051'

**1228789**='0043'

**1229006**='0639'

**1229178**='0622'

**1229704**='0629'

**1259764**='0046'

**1267414**='0036'

**1272512**='0326'

**1302739**='0095'

other = ' ';

**run**;