**5 March 2019**

CDR Chemistry

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 | 05/07/2012 | C. Kangas |  | Baseline |
| 1.01.00 | 12/13/2018 | 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. |

# CDR Chemistry

1. Background

This specification describes the process required to create the MDR Chemistry tables based on data received from the Clinical Data Repository (CDR). The scope of the chemistry lab result data coming from the CDR is a subset of MHS direct care lab orders and results. Separate MDR datasets contain lab result data related to Pathology and Microbiology. The MDR has contained ancillary lab order datasets for many years, generated from extracts received from CHCS. This Chemistry SAS dataset represents the first clinical lab result data available in the MDR.

1. Sources

The source data files used to create the MDR Chemistry table 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 raw extract files are listed below:

**Table 1. CDR Sources**

| **CDR Source** | **Data Files** | **Purpose** |
| --- | --- | --- |
| CDR Lab Orders Table | LBORD\*.DAT | Collection of records for direct care Laboratory Orders in raw text form, prepared in accordance with the ICD. |
| CDR Chemistry Results Summary Table | LBSDS\*.DAT | Collection of records for direct care Chemistry results (summary) in raw text form, prepared in accordance with the ICD. |
| CDR Chemistry Results Detail Table | LBSDD\*.DAT | Collection of records for direct care Chemistry results (detail) 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 Laboratory Orders Table | Weekly |
| CDR Chemistry Results Summary Table | Weekly |
| CDR Chemistry Results Detail 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 is a single MDR Chemistry dataset that is output by the processor:

* The Chemistry table is a single FY level SAS dataset. Each record represents a single result for a lab test. The lab ordering information is contained on each record as well. Many lab tests are ordered as panels (ex. Comprehensive Metabolic Panel), which are groups of chemical tests (ex. Albumin level, Glucose level, etc.) performed on the blood. Therefore, a single test panel can span across many detail result records in this dataset. 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 records across extract cycles.

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 Chemistry Processor Output Products**

|  |  |  |
| --- | --- | --- |
| **Chemistry Processor Output** | **File Naming Convention** | **Member Name** |
| MDR Chemistry File | /mdr/pub/cdr/chem/ | fy<yy>.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 result dates in FY09 or later are kept.
4. Field Transformations and Deletions for MDR
5. The key fields that link the Lab Orders (LBORD\*) feeds to the Lab Results Summary (LBSDS\*) feeds are the HOST\_FACILITY\_ID and LAB\_ORDER\_ID.
6. The key field that links the Lab Results Summary (LBSDS\*) feeds to the Lab Results Detail (LBSDD\*) feeds is the EVENT\_ID field.
7. During the extraction of the raw lab orders records, de-duplication of records, or anytime a lab 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 + LAB\_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 lab orders dataset.

1. During the extraction of the raw lab result summary records, de-duplication of records, or anytime a chemistry result event 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 event key (HOSTDMIS + LAB\_ORDER\_ID + EVENT\_ID). If multiple records exist with the same FEEDDATE and event key, the record with the largest ROW\_NUMBER is kept.
2. During the extraction of the raw lab result detail records, de-duplication of records, or anytime a chemistry result event 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 event key (EVENT\_ID + LAB\_RESULT\_ID). If multiple records exist with the same FEEDDATE and event key, the record with the largest ROW\_NUMBER is kept.
3. There are several merges and formats required to prepare the MDR Chemistry tables as described in Table 4.

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

| **Merge** | **Date Matching** | **Additional Matching** |
| --- | --- | --- |
| CDR Appointment Table | Most recent CDR Appt Table for the FY is used for all FY Chemistry files. | See CDR Appointment Specification. |
| CDR Patient Table | Most recent CDR Patient Table is used for all FY Chemistry files. | See CDR Patient Specification. |
| Longitudinal VM File | Date Resulted, with begin and end dates for each changeable demographic segment | EDI\_PN (if available). |
| CHCS Host Format | None | Apply the format to host\_facility\_id, which will return HOSTDMIS. |
| DMIS ID Format | None | Apply the format to the raw CDR\_CLINIC\_ID, which will return the DMISID. |
| 3M HDD | None | Apply HDD-based formats to lab\_result\_id to obtain LOINC code, LOINC name, and Result Name. |

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 Chemistry 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 Chemistry SAS Data Set

| **Variable Name** | **SAS Field Name** | **Format** | **Source Feeds** | **Related Source Field** | **Transformation Rule** |
| --- | --- | --- | --- | --- | --- |
| CHCS Host DMIS ID | HOSTDMIS | $4. | LBORD, LBSDS | HOST\_FACILITY\_ID | Derived from application of the CDR Host DMISID format: hostdmis = put(host\_facility\_id,hostdmis.);  See Appendix A for the hostdmis format. |
| Lab Order ID | LAB\_ORDER\_ID | $12. | LBORD, LBSDS | LAB\_ORDER\_ID | No transformation. This field is part of the key (along with HOST\_FACILITY\_ID) to join the LBORD and LBSDS feeds together. |
| Treatment DMISID | DMISID | $4. | LBORD | CLINIC\_ID | Derived from application of the CDR DMISID format: DMISID = put(cdr\_clinic\_id,cdr\_dmis.); |
| Lab Test NCID | LAB\_TEST\_NCID | 8. | LBORD, LBSDS | LAB\_TEST\_ID, TEST\_NCID | Derived as TEST\_NCID from LBSDS if not null. If TEST\_NCID is null, populate with LAB\_TEST\_ID from LBORD. |
| Lab Test Name (Raw) | LAB\_TEST\_NAME\_RAW | $30. | LBORD | UNIVERSAL\_SERVICE\_TEXT | No transformation. |
| Lab Test Name (HDD) | LAB\_TEST\_NAME\_HDD | $35. | LBORD, LBSDS | LAB\_TEST\_ID, TEST\_NCID | Derived from application of HDD lab test name format; LAB\_TEST\_NAME\_HDD = put(lab\_test\_ncid,lab\_test\_name.); |
| Order Date | DATE\_ORDERED | $8. | LBSDS | TEXT\_COMMENT | Derived from the TEXT\_COMMENT field, when the field value starts with “DATE\_ORDERED:”. Take following 8 characters. |
| Collect Date | DATE\_COLLECTED | $8. | LBSDS | DATE\_COLLECTED | No transformation. |
| Result Date | DATE\_RESULTED | $8. | LBSDS, LBSDD | TEXT\_COMMENT | Derived from the TEXT\_COMMENT field, when the field value starts with “DATE\_RESULTED:”. Take following 8 characters. |
| Fiscal Year | FY | $4. | LBSDS | DATE\_RESULTED | Fiscal year equivalent of calendar year of DATE\_RESULTED. |
| Fiscal Month | FM | $2. | LBSDS | DATE\_RESULTED | Fiscal month equivalent of calendar month of DATE\_RESULTED. |
| Accession Number | ACCESSNO | $17. | LBSDS | FILLER\_NUMBER | No transformation. |
| Accession System NCID | ACCESS\_SYSTEM\_NCID | 8. | LBSDS | FILLER\_SYSTEM\_NCID | No transformation. |
| Specimen | SPECIMEN | $30. | LBORD | SPECIMEN\_TEXT | No transformation. |
| Specimen Code | SPECIMEN\_CODE | 8. | LBORD | SPECIMEN\_CODE | No transformation. |
| Priority | PRIORITY | $1. | LBORD | PRIORITY | No transformation. |
| Lab Order Grouping | GROUP\_ID | 8. | LBORD | GROUP\_ID | No transformation. |
| Lab Result Event ID | LAB\_EVENT\_ID | 8. | LBSDS, LBSDD | EVENT\_ID | No transformation. This field is the key field to join the LBSDS and LBSDD feeds together. |
| Lab Result NCID | LAB\_RESULT\_NCID | 8. | LBSDD | LAB\_RESULT\_ID | No transformation. |
| Result LOINC | RESULT\_LOINC | $9. | LBSDD | LAB\_RESULT\_ID | Derived from application of HDD LOINC format; RESULT\_LOINC = put(lab\_result\_ncid,loinc.); |
| Result LOINC Name | RESULT\_LOINC\_NAME | $70. | LBSDD | LAB\_RESULT\_ID | Derived from application of HDD LOINC Name format; RESULT\_LOINC\_NAME = put(lab\_result\_ncid,loinc\_name.); |
| Result Name (HDD) | RESULT\_NAME\_HDD | $35. | LBSDD | LAB\_RESULT\_ID | Derived from application of HDD lab test name format; RESULT\_NAME\_HDD = put(lab\_result\_ncid,lab\_test\_name.); |
| Result (Numeric) | RESULT\_NUMERIC | 8. | LBSDD | RESULT\_NUMERIC | No transformation. |
| Result (Text) | RESULT\_TEXT | $40. | LBSDD | RESULT\_TEXT | No transformation. |
| Range | RANGE | $40. | LBSDD | RANGE | No transformation. |
| Units | UNITS | $20. | LBSDD | UNITS\_ID | Derived from application of HDD Units format. UNITS = put(units\_id,units.); |
| Abnormal Indicator | ABNORMAL\_IND | $2. | LBSDD | ABNORMAL\_FLAG\_ID | Derived from application of HDD Abnormal format. ABNORMAL\_IND = put(abnormal\_flag\_id,abnormal.); |
| Result Comment | RESULT\_COMMENT | $100. | LBSDD | TEXT\_COMMENT | Derived from the TEXT\_COMMENT field, when the field value starts with “RESULT COMMENT(S):”. Take following 100 characters. |
| Order Comment | ORDER\_COMMENT | $20. | LBORD | ORDER\_COMMENT | Derived by keeping first 20 characters of the source field ORDER\_COMMENT. |
| Sample Collected | SAMPLE\_COLLECTED | $30. | LBSDS | TEXT\_COMMENT | Derived from the TEXT\_COMMENT field, when the field value starts with “COLLECT\_SAMPLE:”. Take following 30 characters. |
| Order Start Date | DATE\_START | $8. | LBORD | DATE\_START | Derived using first 8 characters of DATE\_START. |
| Order End Date | DATE\_END | $8. | LBORD | DATE\_END | Derived using first 8 characters of DATE\_END. |
| Order Status | ORDER\_STATUS | $2. | LBORD | ORDER\_STATUS\_CODE | No transformation. |
| Provider Name | PROVIDER\_NAME | $40. | LBSDS | TEXT\_COMMENT | Derived from the TEXT\_COMMENT field, when the field value starts with “PROVIDER:”. Take following 40 characters. |
| CDR Patient ID | CDR\_PATIENT\_ID | $20. | LBORD, LBSDS | UNIT\_NUMBER | No transformation. |
| CDR Appointment ID | CDR\_APPT\_ID | 8. | LBORD | APPT\_ID | No transformation. |
| CDR Host ID | CDR\_HOST\_ID | 8. | LBORD, LBSDS | HOST\_FACILITY\_ID | No transformation. This is part of the key (along with LAB\_ORDER\_ID) to join the LBORD and LBSDS feeds together. |
| CDR Clinic ID | CDR\_CLINIC\_ID | 8. | LBORD | CLINIC\_ID | No transformation. |
| CDR Provider ID | CDR\_PROVIDER\_ID | 8. | LBORD | CLINIC\_ID | No transformation. |
| Feed Date | FEEDDT | $8. | N/A | N/A | 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 Appt Table Merge** | | | | | |
| Appt ID Number | APPTIDNO | $10. | N/A | N/A | Derived from the Appt table merge based on CDR\_APPT\_ID |
| **CDR Patient Table Merge** | | | | | |
| Universal Patient ID | UPID | $14. | N/A | N/A | Derived from the Patient table merge based on CDR\_PATIENT\_ID |
| EDIPN | EDI\_PN | $10. | N/A | N/A | Derived from the Patient table merge based on CDR\_PATIENT\_ID |
| Patient SSN | PATSSN | $9. | N/A | N/A | Derived from the Patient table merge based on CDR\_PATIENT\_ID |
| Sponsor SSN | SPONSSN | $9. | N/A | N/A | Derived from the Patient table merge based on CDR\_PATIENT\_ID |
| Patient Date of Birth | PATDOB | 8. | N/A | N/A | Derived from the Patient table merge based on CDR\_PATIENT\_ID |
| Patient Age | PATAGE | 8. | N/A | N/A | Derived using aprod/util macro by subtracting PATDOB from Date Start |
| Patient Category | PATCAT | $3. | N/A | N/A | Derived from the Patient table merge based on CDR\_PATIENT\_ID |
| Age Group Code | AGEGRP | $1. | N/A | N/A | Derived using aprod/util macro. A = 0-17, B = 18-24, etc. |
| **LVM Table Merge** | | | | | |
| DEERS Gender | GENDER | $1. | N/A | N/A | Fill with gender from LVM based on EDIPN. If the gender is blank or U, set to “Z”. |
| DEERS Enrollment DMISID | DENRSITE | $4. | N/A | N/A | 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. | N/A | N/A | 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. | N/A | N/A | 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. | N/A | N/A | 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. | N/A | N/A | 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. | N/A | N/A | 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. | N/A | N/A | 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 HCDP | HCDP | $3. | N/A | N/A | 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. | N/A | N/A | 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. | N/A | N/A | 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. | N/A | N/A | 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. | N/A | N/A | 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. | N/A | N/A | 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. | N/A | N/A | 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. |

1. Refresh Frequency

Frequency of updates (based on Chemistry Result 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 Chemistry 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 the Result 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, LAB\_RESULT\_ID, and DMISID should be tracked across monthly updates.
* When reading in the Chemistry data feeds, a small number of records should be proc printed 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 Chemistry 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**;