



MDS 3.0 Quality Measures

DRAFT USER'S MANUAL Part 2

MDS 3.0 Episode and Stay Determination Logic

(v1 5-20-2011)
DRAFT

Prepared for:
The Centers for Medicare & Medicaid Services
under Contract No. HSM-500-2008-00021I.
(RTI Project Number 0211942.001.100.004)

MDS 3.0 Episode and Stay Determination Logic

Introduction

Several CMS applications are based upon the identification of stays and episodes using MDS 3.0 data. This document provides definitions and detailed logic that can be used by these applications.

This document begins with definitions of key terms and concepts. It then explains how stays and episodes are identified in a well-defined assessment data stream (i.e., when all assessment completion and submission rules are followed). It concludes with detailed logic that handles exceptional cases (e.g., missing entry or discharge records).

Definitions

An episode consists of one or more stays, and a stay is defined as a set of contiguous days in a facility. Because an episode is built from a set of one or more stays, the episode can be identified if the stays have been built properly. Therefore, this section will describe how to build stays.

Three properties of each stay must be determined:

- The starting date.
- The ending date.
- The stay type (admission or reentry).

The starting date is the date the resident entered the facility (either for the first time or after a previous discharge). The ending date is either (a) the discharge date, or (b) the end of the target period, whichever is earlier. The stay type is defined as follows:

Admission. An admission occurs when *any one* of the following conditions apply:

- the resident has never been admitted to this facility before; OR
- the resident has been in this facility previously and was discharged return not anticipated; OR
- the resident has been in this facility previously and was discharged return anticipated and did not return within 30 days of discharge.

Reentry. A reentry occurs when *both of the following* conditions apply:

- the resident has a discharge return anticipated, AND
- the resident returned to the facility within 30 days of discharge.

Rules for a Well Constructed Data Stream

In a well constructed data stream (where all records are submitted and correctly coded), the following logic will correctly determine the starting date, ending date, and type for each stay. This logic assumes that the resident's records have been sorted in reverse chronological order (see the end of this section for sorting details). Stays and episodes must be contained within a single facility, so the following logic applies to the records for a single facility.

1. If the first (latest) record that is on or before the end of the reporting period is a discharge (A0310F = [10, 11, 12]), then the **stay end date** is equal to the discharge date (A2000). Otherwise, the stay is ongoing and the **stay end date** is equal to the end of the reporting period.
2. If the **stay end date** of the resident's latest stay chronologically precedes the beginning of the target period¹, then the episode is not included in the sample. If the stay is ongoing or if the discharge occurs within the target period, then continue.
3. Scan backwards chronologically until an entry record (A0310F = [01]) is encountered. The **stay start date** is equal to the entry date (A1600) on the entry record.
4. Look at the chronologically preceding record. The stay type is defined as follows:
 - 4.1. If a chronologically preceding record is found and if it is a discharge return anticipated (A0310F = [11]) and if the discharge date of the discharge record is within 30 days of the stay start date defined above, then the stay type is a reentry. Otherwise, the stay type is an admission. Admissions occur under any of the following conditions:
 - 4.1.1. No chronologically preceding record is found.
 - 4.1.2. A chronologically preceding record is found and it is a discharge return not anticipated (A0310F = 10).
 - 4.1.3. A chronologically preceding record is found and it is a discharge return anticipated (A0310F = 11) and the discharge date is 31 days or more before the stay start date.
5. If the stay was classified as an admission stay, then scanning would stop because this would mark the beginning of the episode. If the stay was a reentry, then the scan logic would continue with the stay that ended with the record found in Step #4 (if any). Stays would continue to be scanned and classified until one of the following conditions occurred:
 - 5.1. An admission stay was identified, or
 - 5.2. No more records were found for the same resident and facility, or

¹ The span of time that defines the application's reporting period (e.g., a calendar quarter).

- 5.3. An application-specific rule was met. For example, for Quality Measures (QMs), processing stops when the number of cumulative days in the facility (CDIF) exceeded 100 days (CDIF is the sum of the number of days within each of the stays that are contained in the episode).

Handling Missing Records

Exceptions to the rules will occur when entry and/or discharge records are missing from a resident's data stream. When this occurs, starting and/or ending dates must be imputed and the stay type must be determined as accurately as possible. The following rules will describe how these situations are handling. This discussion will refer to three types of records:

- Entry record (where A0310F = [01]).
- Discharge record (where A0310F = [10, 11, 12]).
- A normal assessment (where A0310F = [99]).

Missing Entry Records

In the scan logic described above, if a normal assessment is immediately preceded chronologically by a discharge record or if there is no chronologically preceding record, then an entry record is missing. In this case the stay start date and type must be imputed. The imputation rules are as explained below. In these rules, the assessment that is preceded chronologically by a discharge or that has no preceding record is termed the "problem assessment".

The table below is used to impute the entry date when there is a missing entry record.

Type of Assessment	Reasons for Assessment	Possible Entry Dates	
		Earliest Date	Latest Date
5-day PPS	A0310B=[01]	A2300 - 7 days	A2300
14-day PPS	A0310B=[02]	A2300 - 18 days	A2300 - 10 days
30-day PPS	A0310B=[03]	A2300 - 33 days	A2300 - 20 days
60-day PPS	A0310B=[04]	A2300 - 63 days	A2300 - 49 days
90-day PPS	A0310B=[05]	A2300 - 93 days	A2300 - 79 days
PPS readmission/ return	A0310B=[06]	A2300 - 7 days	A2300
OBRA admission	A0310A=[01]	A2300 - 13 days	A2300
Other OBRA	A0310A=[02,03,04,05,06]	A2300 - 106 days	A2300
OMRA	A0310B=[07]	A2300 - 7 days	A2300
Discharge	A0310F=[10,11,12]	A1600	A1600

The table above lists various types of assessments and shows the earliest and latest possible entry dates that are associated with each one. The following steps explain how to use this table to impute an entry date and stay type when a problem assessment is chronologically preceded by a discharge assessment or where no record precedes the problem assessment.

1. Use the table above to classify the problem assessment. Classify the assessment using the reason for assessment items indicated in the table. If the problem assessment qualifies for more than one of the rows in the table, use the first (topmost) row for which it qualifies.
2. Determine the earliest and latest entry date associated with the selected row.
3. Determine the entry date (A1600) that is reported on the problem assessment.

4. Determine a tentative entry date, as follows:
 - 4.1. If the entry date (A1600) on the problem assessment falls between the earliest and latest entry date in the table, set the tentative entry date equal to this value of A1600.
 - 4.2. Otherwise, set the tentative entry date equal to the date that is listed in the “earliest date” column of the table.
5. Determine a final imputed entry date, as follows:
 - 5.1. If the problem assessment is chronologically preceded by a discharge record, add one day to the tentative entry date and compare the resulting entry date with the discharge date (A2000) on the discharge record. Set the final imputed entry date equal to the later of these two dates.
 - 5.2. If there is no record that chronologically precedes the problem assessment, then set the final imputed entry date equal to the tentative entry date.
6. Determine the stay type, as follows:
 - 6.1. If the problem assessment is chronologically preceded by a discharge record, determine the stay type using the normal logic described above.
 - 6.2. If there is no record that chronologically precedes the problem assessment, then set the stay type as an admission stay.

Missing Discharge Records

In the scan logic described above, if an entry record is immediately preceded chronologically by a normal assessment, then a discharge record is missing. In this case, the end date of the chronologically preceding stay and the stay type of the current stay must be imputed. The imputation rules are as follows. In these rules, the assessment that chronologically precedes the entry record is termed the “ending index assessment”. The “current stay” is the stay that begins with the entry record. The “chronologically preceding stay” is the stay that contains the ending index assessment.

1. The end date of the chronologically preceding stay is set equal to the assessment reference date that is recorded on the ending index assessment.
2. Set the stay type of the current stay as follows:
 - 2.1. Determine the value of A1700 that is recorded on the entry record of the current stay.
 - 2.2. If A1700 is equal to [1] (admission), then set the stay type for the current stay to “admission”.
 - 2.3. If A1700 is equal to [2] (reentry), then set the stay type for the current stay to “reentry”.

Multiple Entry Records

If there are two or more entry records which are adjacent to one another in the resident's data stream, keep the latest entry record and ignore the earlier adjacent entry record(s).

Multiple Discharge Records

If there are two or more discharge records which are adjacent to one another in the resident's data stream, keep the latest discharge record and ignore the earlier adjacent discharge record(s).

Sorting Rules

As noted above, stays are identified from the records for a given resident and facility that are sorted in reverse chronological order. Sorting criteria must be applied to handle the case where there is more than one record on a given target date. The exact sorting criteria are as follows:

State ID +
Facility internal ID +
Resident internal ID +
Target date (descending) +
Record type (descending) +
Assessment internal ID (descending)

Note that record type (record_type) is defined as follows:

1. If A0310F = 01 (the record is an entry record), then record_type = 1.
2. Else if A0310F = 99 (the record is not an entry or discharge), then:
 - a. If the item subset code² is equal to NC (comprehensive assessment), then record_type = [7].
 - b. Else if the item subset code is equal to NQ (quarterly assessment), then record_type = [6].
 - c. Else if the item subset code is equal to NP (PPS assessment), then record_type = [5].
 - d. Else if the item subset code is equal to NO ("other" OMRA assessment), then record type = [4].
 - e. Else if the item subset code is equal to NS (start-of-therapy OMRA assessment), then record_type = [3].
 - f. Else record_type = [2] (this condition should not occur).
3. Else if A0310F = [10] (discharge, return not anticipated), then record_type = [8].
4. Else if A0310F = [11] (discharge, return anticipated), then record_type = [9].
5. Else if A0310F = [12] (death in facility), then record_type = [10].

² The item subset code is contained in the field ITM_SBST_CD.

Also note that the assessment internal ID is used as the final tie-breaker on the assumption that records that should be later in the sort sequence will be submitted and processed later than the other records. The record processing timestamp would be a slightly better field to use for this purpose. However, it is available only to users who have direct access to the ASAP database. The assessment internal ID was therefore adopted as a reasonable substitute for the timestamp so that all users would have access to the same sorting fields.