

999 MISSING SEGMENT / DATA ELEMENTS

RFI 2008-05, JUNE 2009

REQUEST

For the 999 transaction, as a result of routine testing and/or when a WEDI Level 6 or WEDI Level 4 testing error is determined to exist, how is a segment or data element that is *missing* identified? The 'error' codes being used are:

data element IK304, code list 620:

code value 3: Required Segment Missing

code value I6: Implementation Dependent Segment Missing

data element IK403, code list 621:

code value 1: Required Data Element Missing

code value 2: Conditional Required Data Element Missing

code value I9: Implementation Dependent Data Element Missing.

Since the segments and/or data elements are *missing*, no references to them within the received transactions being acknowledged are possible as the precise segment or data element sequence numbers don't exist.

I have searched TR3 005010X231 and can find no discussions or explanations of how to identify *missing* segments or data elements.

And, unfortunately, the examples on pages 47-48 are not sufficiently edifying.

For reference, WEDI Level 6 testing is:

"The specialized testing required by certain healthcare specialties. For example, ambulance, chiropractic, podiatry, home health, parenteral and enteral nutrition, durable medical equipment, psychiatry, and other specialties have specific requirements that must be tested before putting the transaction into production."

WEDI Level 4 testing is:

"The testing of specific inter segment situations described in the HIPAA implementation guides, such that: If A occurs then B must be populated. This is considered to include the validation of situational fields given values or situations present elsewhere in the file. Example: if the claim is for an accident, the accident date must be present."

REFERENCED X12 STANDARDS

A "Request for Interpretation" applies to a specific version of the X12 Standards. The author failed to provide a specific version of the standard in the request. We have chosen to base this response on Version 5 Release 5 of the X12 Standard. As the areas of the X12 Standards applicable to this interpretation have been relatively stable over time, it is likely that the same interpretation would be provided for earlier versions of the X12 Standards.

The following X12 Standards were reviewed in developing this interpretation:

| | | |
|-------------------------------------|-------------|-------------|
| X12.1 Transaction Sets | Version 005 | Release 010 |
| X12.6 Application Control Structure | Version 005 | Release 010 |
| X12.22 Data Segment Directory | Version 005 | Release 010 |
| X12.3 Data Element Dictionary | Version 005 | Release 010 |

INTERPRETATION

The ASC X12 standards do not specify algorithms for identifying missing segments or data elements in a transaction set instance. The 999 Transaction Set provides mechanisms for reporting errors.

In the cases of a required or implementation dependent segment being missing from a transaction set instance, the missing segment is not specifically identified. The data elements of the IK3 segment are populated as follows:

- IK301 – DE721 – Segment ID Code – Segment ID of the segment where the missing segment error was detected
- IK302 – DE719 – Segment Position in Transaction Set – Position in the transaction set of the segment where the missing segment error was detected
- IK304 – DE620 - Implementation Segment Syntax Error Code – Value of “7” or “16”

In the cases of a required, conditional required, or implementation dependent data element being missing from a transaction set instance, the data elements of the IK4 segment are populated as follows:

- IK401 – C030 – Position in Segment – The position of the data element in error within the segment, the composite data structure, or series of repeating data elements, within the segment.
- IK402 – DE725 – Data Element Reference Number – The reference number of the missing data element
- IK403 – DE621 - Implementation Data Element Syntax Error Code – Value of “1”, “2”, or “19”

FURTHER DISCUSSION

As noted in the formal interpretation, ASC X12 does not specify algorithms for validating segment sequence and similar parsing problems. However, we note that in industry practice different algorithms may provide varying degrees of insight into the root cause of a sequence error.

For example, let us consider TS102 in version 005010:

Table 1

| | pos.no | seg.id | name | req.des | max | loop.rep |
|---|--------|--------|---------------------------------------|---------|-----|----------|
| | 0100 | ST | Transaction Set Header | M | 1 | |
| N | 0200 | ORI | Object Reference Identification | M | 1 | |
| N | 0300 | REF | Reference Information | O | >1 | |
| N | 0400 | OOI | Associated Object Type Identification | M | >1 | |
| N | 0500 | BDS | Binary Data Structure | M | 1 | |
| | 0600 | SE | Transaction Set Trailer | M | 1 | |

Let us consider this example

```
ST*102*0001~  
BDS*ASC*1*A~  
SE*3*0001~
```

In this example the ORI and OOI segments, which are mandatory in the standard, are absent. At the BDS segment, the following conditions are true:

- The BDS segment is unexpected after the ST segment
- The BDS segment is not in proper sequence
- The mandatory ORI segment is missing
- The mandatory OOI segment is missing

A report indicating any of these conditions as an error would be correct. Different algorithms may put these errors in different precedence, or may not implement all of them.