

OFFICIAL

Elective Surgery Waiting List

Data Quality Checks

Reference Manual
2019-2020

OFFICIAL



OFFICIAL



Government
of South Australia

SA Health

For Official Use Only-I1-A1

OFFICIAL

Table of Contents

Document Control Information	4
Change History	4
Approvals.....	5
Document Endorsement	5
Executive Summary	6
Changes	6
Additional Information	6
Data Quality Checks	7
Improving Errors	7
Data Quality Checks	7
11: Invalid Sex Code.....	8
12: INVALID POST CODE.....	9
13: Invalid Birth Date	10
14: Invalid Election Code	11
15: Invalid Procedure Code	12
16: Invalid Status Code.....	14
17: Invalid Unit Code	15
17: Warning Unexpected Unit Code	15
18: Invalid Date Added	16
19: Date Added Prior To Birth Date	17
20: Invalid Admit/Remove Date	18
21: Admit/Remove Date prior TO Date Added	19
22: Invalid Priority Code.....	20
23: Treatment Records: Xmitted	21
24: Invalid Change Date	22
25: Invalid Change Code	23
26: Cancellation Records: Xmitted	25
27: Invalid Category Assessment Date.....	26
28: Invalid Effective Date	27
29: Invalid Old Category	28
29: Invalid Old Category	29
30: Invalid New Category.....	30
30: Invalid New Category.....	31
32: Invalid Same Day Code.....	32
32: Invalid Same Day Code.....	33

33: Status is Not Removed;Invalid Reason	34
33: Status is Removed;Invalid Reason.....	34
34: Invalid Patient Unit Number	36
35: Change Date earlier than date Added	37
35: Change Date later than date Removed	38
36: Invalid Indigenous Status.....	39
37: Invalid Blank Deferral Reason	40
37: Invalid Deferral Reason	40
38: Empty Scheduled Admission Date For Status.....	42
39: Scheduled Admission Date Before Date on List.....	43
40: Empty Booking Date For Status	44
41: Booking Date Before Date on List.....	45
42: Main CMBS Code Invalid.....	46
43: Main CMBS Code Excluded	48
44: No CMBS Code Provided	50
45: Deferred Admission (Cat 4) With Treatment Status.....	52
45: Removal Reason Emergency (03) and Deferred Admisson (Cat 4) With Treatment Status ...	52
46: Admit/Remove Date Prior TO Date Created.....	53
47: Excluded Legacy Indicator Procedure Code.....	54
48: Excluded Intended Procedure Code For Past Record.....	56

Document Control Information

Title: Elective Surgery Waiting List - File Extract - Technical Specification
2019-2020

Description: This document describes the file submission format used in the Elective Surgery Waiting List data collection.

Version 1.0

Version Date 19-Sep-2019

Document Owner: Manager, Data Governance and Integrity

Contributors: Damien Bourke: Data Steward Team Leader
Georgia Lefty: Senior Project Officer

Document Location: \\dhgsf02\ims\Knowledge Management\Enterprise Information Services\CDCS\1 July Process\2019\04 Elective Surgery Waiting List\06 Manual

Document Classification For Official Use Only-I1-A1

Change History

A printed version of this document may have been superseded. The current version of this document can be accessed via the project site document library in Project Server.

Document Version	Date	Updated By	Change Summary
0.1	01-Apr-2019	Georgia Lefty: Senior Project Officer	Initial version
0.2	01-May-2019	Damien Bourke: Data Steward Team Leader	Internal review
0.3	08-Jul-2019	Susan Sander: Senior Manager, Corporate Data Collection Systems	Internal review
1.0	01-Sep-2019	Damien Bourke: Metadata and Change Manager	Internal review

Approvals

This document is approved on the basis that it meets the following acceptance criteria.

Document Endorsement

This document requires the following endorsement:

Version	Date	Name	Endorsed Yes/No?	Signature
1.0	19-Sep-2019	Susan Sander: Senior Manager, Corporate Data Collection Systems	Yes	SQS

Executive Summary

Changes

Standardised format for pseudocode e.g.

```
IF [Data element] (data item number) = something invalid
THEN
    ACCEPT submitted record
    PRINT error message xxxx: description
```

Additional Information

Clinical unit codes are described in the spreadsheet **Elective Surgery Waiting List - Reference File - Clinical Units**.

Hospital codes are described in the spreadsheet **Elective Surgery Waiting List - Reference File - Hospital Codes**.

Intended procedure codes are described in the spreadsheet **Elective Surgery Waiting List - Reference File - Intended Procedure**.

Additional resources are available from:

<https://www.sahealth.sa.gov.au/wps/wcm/connect/public+content/sa+health+internet/about+us/our+performance/elective+surgery+waiting+list/elective+surgery+waiting+list+eswl+resources>

Data Quality Checks

Improving Errors

If you suspect that a data quality check is incorrect, or raising a large number of queries unnecessarily, please contact the BLIS Unit through the DAI service management to raise your concerns

Data Quality Checks

A data quality check is produced for each hospital once the sites data has been loaded into the BLIS database. The report is emailed to each hospital who then forward the corrections back to the Department and corrections are made in the database.

Data quality checks are explained in the section that follows.

11: Invalid Sex Code**Pseudocode**

```

IF
(
    [Sex] (data item 25) ≠ M
    AND
    [Sex] (data item 25) ≠ F
)
THEN
    PRINT error message 11: Invalid Sex Code

```

Status

Active.

Implementation

```

IF .not. xsex$'MF'
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH urnum + "      Invalid Sex Code - " + temptr->xsex
    REPLACE errnumber WITH "11"
    SELECT 2
ELSE
    IF xsex$'M'
        STORE ctm + 1 TO ctm
    ELSE
        STORE ctf + 1 TO ctf
    ENDIF
ENDIF
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

12: INVALID POST CODE**Pseudocode**

```

IF NOT
(
    [Postcode] (data item 20) ≥ 0800 and ≤ 0899
    OR
    [Postcode] (data item 20) ≥ 2000 and ≤ 7999
    OR
    [Postcode] (data item 20) = 9999
)
THEN
    PRINT error message 12: INVALID POST CODE

```

Status

Active.

Implementation

```

IF (val(xpcode) >= 0800 .and. val(xpcode)<=0899).or.(val(xpcode) >= 2000 .and.
val(xpcode)<=7999).or.xpcode='9999'
    * nothing
ELSE
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH urnum +"          INVALID POST CODE - " + TEMPTR->XPCODE
    REPLACE errnumber WITH "12"
    SELECT 2
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

13: Invalid Birth Date**Pseudocode**

```

IF
(
    [Date Of Birth] (data item 12) is invalid
)
THEN
    PRINT error message 13: Invalid Birth Date

```

Status

Active.

Implementation

```

IF xdob <> ctod(dtoc(xdob))
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH urnum + "      Invalid Birth Date - " + dtoc(temptr->xdob)
    REPLACE errnumber WITH "13"
    SELECT 2
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

14: Invalid Election Code**Pseudocode**

```

IF
(
    [Election] (data item 15) <> H or P
)
THEN
    PRINT error message 14: Invalid Election Code

```

Status

Active.

Implementation

```

IF .not. xselect$'HP'
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH urnum + "      Invalid Election Code - " + temptr->xselect
    REPLACE errnumber WITH "14"
    SELECT 2
ELSE
    IF xselect$'H'
        STORE cth + 1 TO cth
    ELSE
        STORE ctp + 1 TO ctp
    ENDIF
ENDIF
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

15: Invalid Procedure Code

Pseudocode

```
IF
(
[Intended Procedure] (data item 17) is invalid
)
THEN
PRINT error message 15: Invalid Procedure Code
```

Status

Active.

Implementation

```
* check for invalid procedure code
STORE "J" + xcode TO gkey
SELECT 5
FIND &gkey
IF eof()
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH urnum + "      Invalid Procedure Code - " + temptr->xcode
    REPLACE errnumber WITH "15"
ELSE
    * note genend part of or section probably not needed
    IF (temptr->xdaton>genend .OR. (temptr->xstatus$'12' .AND. DATE()>=CTOD("01/08/2016")))
.AND. dtoc(genend) <> " / / "
        * old code used
        STORE .T. TO errupd
        SELECT 4
        APPEND BLANK
        REPLACE errtext WITH urnum + "      Excluded Legacy Indicator Procedure Code - " +
temptr->xcode
        REPLACE errnumber WITH "47"
    ELSE
        SELECT 5
        IF temptr->xenddat>genend .AND. dtoc(genend) <> " / / " .AND. dtoc(temptr->xenddat)
<> " / / "
            * old code used too late
            STORE .T. TO errupd
            SELECT 4
            APPEND BLANK
            REPLACE errtext WITH urnum + "      Excluded Legacy Indicator Procedure Code - " +
temptr->xcode
            REPLACE errnumber WITH "47"
        ENDIF
    ENDIF
    SELECT 5
    IF temptr->xenddat<genstart .AND. dtoc(temptr->xenddat) <> " / / "
        * new code used too early
```

```
STORE .T. TO errupd
SELECT 4
APPEND BLANK
REPLACE errtext WITH urnum + "      Excluded Intended Procedure Code For Past Record - "
+ temptr->xcode
REPLACE errnumber WITH "48"
ENDIF
ENDIF
```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

16: Invalid Status Code**Pseudocode**

```

IF
(
    [Treatment Status] (data item 27) is invalid
)
THEN
    PRINT error message 16: Invalid Status Code

```

Status

Active.

Implementation

```

IF eof()
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH urnum + "      Invalid Status Code - " + temptr->xstatus
    REPLACE errnumber WITH "16"
ELSE
    DO CASE
        CASE temptr->xstatus = "1"
            STORE cts1 + 1 TO cts1
        CASE temptr->xstatus = "2"
            STORE cts2 + 1 TO cts2
        CASE temptr->xstatus = "3"
            STORE cts3 + 1 TO cts3
        CASE temptr->xstatus = "4"
            STORE cts4 + 1 TO cts4
        CASE temptr->xstatus = "5"
            STORE cts5 + 1 TO cts5
    ENDCASE
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

17: Invalid Unit Code**17: Warning Unexpected Unit Code****Pseudocode**

```

IF
(
    [Clinical Unit] (data item 9) is invalid
)
THEN
    PRINT error message 17: Invalid Unit Code

IF
(
    [Clinical Unit] (data item 9) is unexpected
)
THEN
    PRINT error message 17: Warning Unexpected Unit Code

```

Status

Active.

Implementation

```

IF eof()
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH urnum + "    Invalid Unit Code - " + temptr->xunit
    REPLACE errnumber WITH "17"
ELSE
    IF genattr="N"
        * warning update only - not error
        SELECT 4
        APPEND BLANK
        REPLACE errtext WITH urnum + "    Warning Unexpected Unit Code - " + temptr->xunit
        REPLACE errnumber WITH "17"
    ENDIF
ENDIF
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

18: Invalid Date Added**Pseudocode**

```

IF
(
    [Date Added To List] (data item 11) is invalid
)
THEN
    PRINT error message 18: Invalid Date Added

```

Status

Active.

Implementation

```

IF xdaton <> ctod(dtoc(xdaton))
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH urnum + "      Invalid Date Added - " + dtoc(temptr->xdaton)
    REPLACE errnumber WITH "18"
    SELECT 2
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

19: Date Added Prior To Birth Date**Pseudocode**

```

IF
(
    [Date Added To List] (data item 11) is < [Date Of Birth] (data item 12)
)
THEN
    PRINT error message 19: Date Added Prior to Birth Date

```

Status

Active.

Implementation

```

IF xdaton < xdob
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH urnum + "    Date Added prior TO Birth Date - " + dtoc(temptr->xdaton) + " < " + dtoc(temptr->xdob)
    REPLACE errnumber WITH "19"
    SELECT 2
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

20: Invalid Admit/Remove Date**Pseudocode**

```

IF
(
    [Treatment Status] (data item 27) = 3, 4, or 5
    AND
    [Admission/Removal Date] (data item 2) is invalid
)
OR
(
    [Treatment Status] (data item 27) <> 3, 4, or 5
    AND
    [Admission/Removal Date] (data item 2) is not invalid
)
THEN
    PRINT error message 20: Invalid Admit/Remove Date

```

Status

Active.

Implementation

```

IF xstatus$('345'.and.(dtoc(temptr->xenddat)=" / /      ".or. temptr->xenddat <>
ctod(dtoc(temptr->xenddat)))
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH urnum +"      Invalid Admit/Remove Date - " + dtoc(temptr->xenddat)
    REPLACE errnumber WITH "20"
    SELECT 2
ENDIF
IF .not. xstatus$('345' .and. dtoc(xenddat) <> " / /      "
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH urnum +"      Invalid Admit/Remove Date - " + dtoc(temptr->xenddat)
    REPLACE errnumber WITH "20"
    SELECT 2
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

21: Admit/Remove Date prior TO Date Added**Pseudocode**

```

IF
(
    [Admission/Removal Date] (data item 2) is < [Date Added To List] (data item 11)
)
THEN
    PRINT error message 21: Admit/Remove Date prior TO Date Added

```

Status

Active.

Implementation

```

IF xenddat < xdaton .and. dtoc(xenddat) <> " / / "
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH urnum + "      Admit/Remove Date prior TO Date Added - " +
dtoc(temptr->xenddat) + " < " + dtoc(temptr->xdaton)
    REPLACE errnumber WITH "21"
    SELECT 2
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

22: Invalid Priority Code**Pseudocode**

```

IF
(
    [Admission Category] (data item 1) is invalid
)
THEN
    PRINT error message 22: Invalid Priority Code

```

Status

Active.

Implementation

```

IF eof()
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH urnum + "        Invalid Priority Code - " + temptr->xurgst
    REPLACE errnumber WITH "22"
ELSE
    DO CASE
        CASE temptr->xurgst = "1"
            STORE ctp1 + 1 TO ctp1
        CASE temptr->xurgst = "2"
            STORE ctp2 + 1 TO ctp2
        CASE temptr->xurgst = "3"
            STORE ctp3 + 1 TO ctp3
        CASE temptr->xurgst = "9"
            STORE ctp4 + 1 TO ctp4
        CASE temptr->xurgst = "4"
            STORE ctp5 + 1 TO ctp5
        CASE temptr->xurgst = "5"
            STORE ctp6 + 1 TO ctp6
    ENDCASE
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

23: Treatment Records: Xmitted**Pseudocode**

```
Count the number of Treatment Records transmitted
THEN
    PRINT error message 23: Treatment Records: Xmitted
```

Status

Active.

Implementation

```
IF rect <> counter1
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH "          Treatment Records: Xmitted = "+str(rect,5,0)+" Copied &
Validated = "+str(counter1,5,0)
    REPLACE errnumber WITH "23"
    SELECT 2
ENDIF
```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

24: Invalid Change Date**Pseudocode**

```

IF
(
    [Effective Date] (data item 14) is invalid
)
THEN
    PRINT error message 24: Invalid Change Date

```

Status

Active.

Implementation

```

IF ycandat <> ctod(dtoc(ycandat))
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH urnum + "      Invalid Change Date - " + dtoc(tempca->ycandat)
    REPLACE errnumber WITH "24"
    SELECT 3
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

25: Invalid Change Code**Pseudocode**

```

IF
(
    [Cancellation Type] (data item 7) is invalid
)
THEN
    PRINT error message 25: Invalid Change Code

```

Status

Active.

Implementation

```

IF eof()
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH urnum + "        Invalid Change Code - " + tempca->ycantyp
    REPLACE errnumber WITH "25"
ELSE
    DO CASE
        CASE tempca->ycantyp = "01"
            STORE ctc1 + 1 TO ctc1
        CASE tempca->ycantyp = "02"
            STORE ctc2 + 1 TO ctc2
        CASE tempca->ycantyp = "03"
            STORE ctc3 + 1 TO ctc3
        CASE tempca->ycantyp = "04"
            STORE ctc4 + 1 TO ctc4
        CASE tempca->ycantyp = "05"
            STORE ctc5 + 1 TO ctc5
        CASE tempca->ycantyp = "06"
            STORE ctc6 + 1 TO ctc6
        CASE tempca->ycantyp = "07"
            STORE ctc7 + 1 TO ctc7
        CASE tempca->ycantyp = "98"
            STORE ctc98 + 1 TO ctc98
        CASE tempca->ycantyp = "99"
            STORE ctc99 + 1 TO ctc99
    ENDCASE
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

26: Cancellation Records: Xmitted**Pseudocode**

```
Count the number of Cancellation Records transmitted
THEN
    PRINT error message 26: Treatment Records: Xmitted
```

Status

Active.

Implementation

```
IF recc <> counter2
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH "          Cancellation Records: Xmitted = "+str(recc,5,0)+" Copied
& Validated = "+str(counter2,5,0)
    REPLACE errnumber WITH "26"
    SELECT 3
ENDIF
```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

27: Invalid Category Assessment Date**Pseudocode**

```

IF
(
    [Treatment Status] (data item 27) = 1
    AND
    [Admission Category] (data item 1) = 4 or 5
    AND
    [Category Assessment Date] (data item 8) is invalid
)
THEN
    PRINT error message 27: Invalid Category Assessment Date

```

Status

Active.

Implementation

```

IF xstatus='1' .and. xurgst$'45' .and. dtoc(xassdat)=' / / '
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH urnum + "      Invalid Category Assessment Date - " + dtoc(temptr-
>xassdat)
    REPLACE errnumber WITH "27"
    SELECT 2
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

28: Invalid Effective Date**Pseudocode**

```

IF
(
    [Category Change Date] (data item 14) is invalid
)
THEN
    PRINT error message 28: Invalid Effective Date

```

Status

Active.

Implementation

```

IF wcatdate <> ctod(dtoc(wcatdate))
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH urnum + "      Invalid Effective Date - " + dtoc(temphis->wcatdate)
    REPLACE errnumber WITH "28"
    SELECT 6
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

29: Invalid Old Category**Pseudocode**

```

IF
(
  [Admission Category] (data item 1) = 5
  AND
  [Category Change Date] (data item 14) ≥ 01/07/2010
)
THEN
  PRINT error message 29: Invalid Old Category

```

Status

Active.

Implementation

```

IF wcatold='5' .AND. wcatdate>=CTOD("01/07/2010")
  STORE .T. TO errupd
  SELECT 4
  APPEND BLANK
  REPLACE errtext WITH urnum + "      Invalid Old Category - " + temphis->wcatold
  REPLACE errnumber WITH "29"
  SELECT 6
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

29: Invalid Old Category**Pseudocode**

```

IF
(
    [Admission Category] (data item 1) is invalid
)
THEN
    PRINT error message 29: Invalid Old Category

```

Status

Active.

Implementation

```

IF eof()
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH urnum + "        Invalid Old Category - " + temphis->wcatold
    REPLACE errnumber WITH "29"
    SELECT 6
ELSE
    DO CASE
        CASE temphis->wcatold = "1"
            STORE cto1 + 1 TO cto1
        CASE temphis->wcatold = "2"
            STORE cto2 + 1 TO cto2
        CASE temphis->wcatold = "3"
            STORE cto3 + 1 TO cto3
        CASE temphis->wcatold = "4"
            STORE cto4 + 1 TO cto4
        CASE temphis->wcatold = "5"
            STORE cto5 + 1 TO cto5
        CASE temphis->wcatold = "9"
            STORE cto6 + 1 TO cto6
    ENDCASE
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

30: Invalid New Category**Pseudocode**

```

IF
(
    [Admission Category] (data item 1) = 5
    AND
    [Category Change Date] (data item 14) ≥ 01/07/2010
)
THEN
    PRINT error message 30: Invalid New Category

```

Status

Active.

Implementation

```

IF wcatnew='5' .AND. wcatdate>=CTOD("01/07/2010")
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH urnum + "      Invalid New Category - " + temphis->wcatnew
    REPLACE errnumber WITH "30"
    SELECT 6
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

30: Invalid New Category**Pseudocode**

```

IF
(
    [Admission Category] (data item 1) is invalid
)
THEN
    PRINT error message 30: Invalid New Category

```

Status

Active.

Implementation

```

IF eof()
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH urnum + "    Invalid New Category - " + temphis->wcatnew
    REPLACE errnumber WITH "30"
    SELECT 6
ELSE
    DO CASE
        CASE temphis->wcatnew = "1"
            STORE ctn1 + 1 TO ctn1
        CASE temphis->wcatnew = "2"
            STORE ctn2 + 1 TO ctn2
        CASE temphis->wcatnew = "3"
            STORE ctn3 + 1 TO ctn3
        CASE temphis->wcatnew = "4"
            STORE ctn4 + 1 TO ctn4
        CASE temphis->wcatnew = "5"
            STORE ctn5 + 1 TO ctn5
        CASE temphis->wcatnew = "9"
            STORE ctn6 + 1 TO ctn6
    ENDCASE
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

32: Invalid Same Day Code**Pseudocode**

```

IF
(
    [Same Day Flag] (data item 23) is <> Y, N, or U
)
THEN
    PRINT error message 32: Invalid Same Day Code

```

Status

Active.

Implementation

```

IF .not. xtday$'YNU'
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH urnum + "      Invalid Same Day Code - " + temptr->xtday
    REPLACE errnumber WITH "32"
    SELECT 2
ELSE
    DO CASE
        CASE xtday='Y'
            STORE ctd1 + 1 TO ctd1
        CASE xtday='N'
            STORE ctd2 + 1 TO ctd2
        CASE xtday='U'
            STORE ctd3 + 1 TO ctd3
    ENDCASE
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

32: Invalid Same Day Code**Pseudocode**

```

IF
(
    [Same Day Flag] (data item 23) is <> Y, N, or U
)
THEN
    PRINT error message 32: Invalid Same Day Code

```

Status

Active.

Implementation

```

IF .not. xtday$'YNU'
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH urnum + "      Invalid Same Day Code - " + temptr->xtday
    REPLACE errnumber WITH "32"
    SELECT 2
ELSE
    DO CASE
        CASE xtday='Y'
            STORE ctd1 + 1 TO ctd1
        CASE xtday='N'
            STORE ctd2 + 1 TO ctd2
        CASE xtday='U'
            STORE ctd3 + 1 TO ctd3
    ENDCASE
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

33: Status is Not Removed;Invalid Reason**33: Status is Removed;Invalid Reason****Pseudocode**

```

IF
(
    [Treatment Status] (data item 27) = 5
AND
    [Reason For Removal] (data item 21) is invalid
)
THEN
    PRINT error message 33: Status is Removed;Invalid Reason

IF
(
    [Reason For Removal] (data item 21)is invalid
)
THEN
    PRINT error message 33: Status is Not Removed;Invalid Reason

```

Status

Active.

Implementation

```

IF temptr->xstatus = '5'
    STORE "G" + xreasof TO gkey
    SELECT 5
    FIND &gkey
    IF eof()
        STORE .T. TO errupd
        SELECT 4
        APPEND BLANK
        REPLACE errtext WITH urnum + "          Status is Removed;Invalid Reason - " + temptr-
        >xreasof
        REPLACE errnumber WITH "33"
    ELSE
        DO CASE
            CASE temptr->xreasof = "01"
                STORE ctr1 + 1 TO ctr1
            CASE temptr->xreasof = "02"
                STORE ctr2 + 1 TO ctr2
            CASE temptr->xreasof = "03"
                STORE ctr3 + 1 TO ctr3
            CASE temptr->xreasof = "04"
                STORE ctr4 + 1 TO ctr4
            CASE temptr->xreasof = "05"
                STORE ctr5 + 1 TO ctr5
            CASE temptr->xreasof = "06"
                STORE ctr6 + 1 TO ctr6
            CASE temptr->xreasof = "07"
                STORE ctr7 + 1 TO ctr7
            CASE temptr->xreasof = "98"
                STORE ctr98 + 1 TO ctr98
            CASE temptr->xreasof = "99"

```

```

        STORE ctr99 + 1 TO ctr99
    ENDCASE
ENDIF
ELSE
    IF VAL(temptr->Xreasof) <> 0
        STORE .T. TO errupd
        SELECT 4
        APPEND BLANK
        REPLACE errtext WITH urnum + "        Status is Not Removed;Invalid Reason - " + temptr-
>xreasof
        REPLACE errnumber WITH "33"
    ELSE
        STORE ctr00 + 1 TO ctr00
    ENDIF
ENDIF
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

34: Invalid Patient Unit Number**Pseudocode**

```

IF
(
    [Patient Unit Record Number] (data item 19) is invalid
)
THEN
    PRINT error message 34: Invalid Patient Unit Number

```

Status

Active.

Implementation

```

IF m->varunitno = '000000000'
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH urnum + "      Invalid Patient Unit Number - " + temptr->xpatunitno
    REPLACE errnumber WITH "34"
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

35: Change Date earlier than date Added**Pseudocode**

```

IF
(
    [Category Change Date] (data item 14) < [Date Added To List] (data item 11)
)
THEN
    PRINT error message 35: Change Date earlier than date Added

```

Status

Active.

Implementation

```

IF tempthis->wcatdate < temptr->xdaton
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH mnum + "      Change Date earlier than date Added - " + dtoc(tempthis->wcatdate) + " < " + dtoc(temptr->xdaton)
    REPLACE errnumber WITH "35"
    SELECT 6
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

35: Change Date later than date Removed**Pseudocode**

```

IF
(
    [Category Change Date] (data item 14) > [Admission/Removal Date] (data item 2)
)
THEN
    PRINT error message 35: Change Date later than date Removed

```

Status

Active.

Implementation

```

IF tempthis->>wcatdate > temptr->xenddat .AND. .NOT. dtoc(temptr->xenddat)=' / / '
SELECT 4
APPEND BLANK
REPLACE errtext WITH mnum + "          Change Date later than date Removed - " + dtoc(tempthis-
>wcatdate) + " > " + dtoc(temptr->xenddat)
REPLACE errnumber WITH "35"
SELECT 6
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

36: Invalid Indigenous Status**Pseudocode**

```

IF
(
    [Indigenous Status] (data item 18) is invalid
)
THEN
    PRINT error message 36: Invalid Indigenous Status

```

Status

Active.

Implementation

```

IF .NOT. EMPTY(temptr->xIndigStat)
    * check value
    SELECT genpar
    SEEK "K"+temptr->xIndigStat
    IF EOF()
        SELECT 4
        APPEND BLANK
        REPLACE errtext WITH urnum + "    Invalid Indigenous Status - " + temptr->xIndigStat
        REPLACE errnumber WITH "36"
    ENDIF
    SELECT 2
ENDIF
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

37: Invalid Blank Deferral Reason**37: Invalid Deferral Reason****Pseudocode**

```

IF
(
    [Admission Category] (data item 1) = 4
    AND
    [Deferral Reason] (data item 13) is invalid
)
THEN
    PRINT error message 37: Invalid Blank Deferral Reason

IF
(
    [Admission Category] (data item 1) = 4
    AND
    [Deferral Reason] (data item 13) is invalid
)
THEN
    PRINT error message 37: Invalid Deferral Reason

```

Status

Active.

Implementation

```

IF temphis->wcatnew='4'
    STORE "L" + temphis->wdefreason TO gkey
    SELECT 5
    FIND &gkey
    IF eof() .OR. EMPTY(temphis->wdefreason)
        STORE .T. TO errupd
        SELECT 4
        APPEND BLANK
        IF EMPTY(temphis->wdefreason)
            REPLACE errtext WITH urnum + "    Invalid Blank Deferral Reason "
        ELSE
            REPLACE errtext WITH urnum + "    Invalid Deferral Reason - " + temphis->wdefreason
        ENDIF
        REPLACE errnumber WITH "37"
    ENDIF
    SELECT 6
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

38: Empty Scheduled Admission Date For Status**Pseudocode**

```

IF
(
    [Treatment Status] (data item 27) = 2, 3 or 4
    AND
    [Scheduled Admission Date] (data item 24) is invalid
)
THEN
    PRINT error message 38: Empty Scheduled Admission Date for Status

```

Status

Active.

Implementation

```

IF temptr->xstatus$"234" .AND. EMPTY(temptr->xSAdmDate)
    * adm date cannot be empty
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH urnum + "    Empty Scheduled Admission Date For Status="+temptr-
>xstatus
    REPLACE errnumber WITH "38"
    SELECT 2
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

39: Scheduled Admission Date Before Date on List**Pseudocode**

```

IF
(
    [Scheduled Admission Date] (data item 24) < [Date Added To List] (data item 11)
    AND
    [Scheduled Admission Date] (data item 24) is not blank
)
THEN
    PRINT error message 39: Scheduled Admission Date Before Date on List

```

Status

Active.

Implementation

```

IF temptr->xSAdmDate < xdaton .AND. !EMPTY(temptr->xSAdmDate)
    * adm date cannot be less than date on
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errrtext WITH urnum + "    Scheduled Admission Date Before Date on List " +
DTC(temptr->xSAdmDate)
    REPLACE errnumber WITH "39"
    SELECT 2
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

40: Empty Booking Date For Status**Pseudocode**

```

IF
(
    [Treatment Status] (data item 27) = 2, 3 or 4
    AND
    [Booking Date] (data item 5) is blank
)
THEN
    PRINT error message 40: Empty Booking Date For Status

```

Status

Active.

Implementation

```

IF temptr->xstatus$"234" .AND. EMPTY(temptr->xBookDate)
    * booking date cannot be empty
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH urnum + "    Empty Booking Date For Status="+temptr->xstatus
    REPLACE errnumber WITH "40"
    SELECT 2
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

41: Booking Date Before Date on List**Pseudocode**

```

IF
(
    [Booking Date] (data item 5) < [Date Added To List] (data item 11)
    AND
    [Booking Date] (data item 5) is not blank
)
THEN
    PRINT error message 41: Booking Date Before Date on List

```

Status

Active.

Implementation

```

IF temptr->xBookDate < xdaton .AND. !EMPTY(temptr->xBookDate)
    * adm date cannot be less than date on
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errrtext WITH urnum + "      Booking Date Before Date on List " + DTOC(temptr-
    >xBookDate)
    REPLACE errnumber WITH "41"
    SELECT 2
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

42: Main CMBS Code Invalid**Pseudocode**

```

IF
(
    [CMBS] (data item 10) is invalid
)
THEN
    PRINT error message 42: Main CMBS Code Invalid

```

Status

Active.

Implementation

* Check 1st CMBS Code

```

IF recm > 0
    SELECT tempcmbs
    SEEK temptr->xurnum
    IF FOUND()
        SELECT 0
        SELECT cmbscode
        SEEK tempcmbs->vcmbrcode
        IF .NOT. FOUND()
            STORE .T. TO errupd
            SELECT 4
            APPEND BLANK
            REPLACE errtext WITH urnum + "      Main CMBS Code Invalid " + tempcmbs-
>vcmbrcode
            REPLACE errnumber WITH "42"
            counter8 = counter8 +1
            SELECT 2
        ELSE
            * check code not excluded
            IF cmbscode->excluded = "Y"
                STORE .T. TO errupd
                SELECT 4
                APPEND BLANK
                REPLACE errtext WITH urnum + "      Main CMBS Code Excluded " + tempcmbs-
>vcmbrcode
                REPLACE errnumber WITH "43"
                counter8 = counter8 +1
                SELECT 2
            ENDIF
        ENDIF
    * add cmbs record order
    replace tempcmbs->vorder WITH 1
    SELECT tempcmbs
    skip
    lnorder = 2
    DO WHILE !EOF() .AND. tempcmbs->vurnum=temptr->xurnum
        replace tempcmbs->vorder WITH lnorder

```

```

        skip
        lnorder=lnorder+1
    enddo
    counter7 = counter7 +1
ELSE
    * CBMS code is maditory but initially this check will be disabled as some hospitals
will lodge incomplete records
    *STORE .T. TO errupd
    *SELECT 4
    *APPEND BLANK
    *REPLACE errtext WITH urnum +"      No CMBS Code Provided "
    *REPLACE errnumber WITH "44"
    SELECT 2
ENDIF
* delete old codes
SELECT cmbs
SEEK hosp+temptr->xurnum
IF !EOF ()
    DO WHILE cmbs->vpnumber=hosp+temptr->xurnum .AND. !EOF ()
        DELETE
        SKIP
        counter8 = counter8 +1
    ENDDO
ENDIF
SELECT 2
ENDIF
*
SELECT 2
skip
IF errupd
    STORE cterrt + 1 TO cterrt
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

43: Main CMBS Code Excluded**Pseudocode**

```

IF
(
    [CMBS] (data item 10) is excluded
)
THEN
    PRINT error message 43: Main CMBS Code Excluded

```

Status

Active.

Implementation

* Check 1st CMBS Code

```

IF recm > 0
    SELECT tempcmbs
    SEEK temptr->xurnum
    IF FOUND()
        SELECT 0
        SELECT cmbscode
        SEEK tempcmbs->vcmbrcode
        IF .NOT. FOUND()
            STORE .T. TO errupd
            SELECT 4
            APPEND BLANK
            REPLACE errtext WITH urnum + "      Main CMBS Code Invalid " + tempcmbs-
>vcmbrcode

            REPLACE errnumber WITH "42"
            counter8 = counter8 +1
            SELECT 2
        ELSE
            * check code not excluded
            IF cmbscode->excluded = "Y"
                STORE .T. TO errupd
                SELECT 4
                APPEND BLANK
                REPLACE errtext WITH urnum + "      Main CMBS Code Excluded " + tempcmbs-
>vcmbrcode

                REPLACE errnumber WITH "43"
                counter8 = counter8 +1
                SELECT 2
            ENDIF
        ENDIF
        * add cmbs record order
        replace tempcmbs->vorder WITH 1
        SELECT tempcmbs
        skip
        lnorder = 2
        DO WHILE !EOF() .AND. tempcmbs->vurnum=temptr->xurnum
            replace tempcmbs->vorder WITH lnorder

```

```

        skip
        lnorder=lnorder+1
    enddo
    counter7 = counter7 +1
ELSE
    * CBMS code is maditory but initially this check will be disabled as some hospitals
will lodge incomplete records
    *STORE .T. TO errupd
    *SELECT 4
    *APPEND BLANK
    *REPLACE errtext WITH urnum +"      No CMBS Code Provided "
    *REPLACE errnumber WITH "44"
    SELECT 2
ENDIF
* delete old codes
SELECT cmbs
SEEK hosp+temptr->xurnum
IF !EOF ()
    DO WHILE cmbs->vpnumber=hosp+temptr->xurnum .AND. !EOF ()
        DELETE
        SKIP
        counter8 = counter8 +1
    ENDDO
ENDIF
SELECT 2
ENDIF
*
SELECT 2
skip
IF errupd
    STORE cterrt + 1 TO cterrt
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

44: No CMBS Code Provided**Pseudocode**

```

IF
(
    [CMBS] (data item 10) is null
)
THEN
    PRINT error message 44: No CMBS Code Provided

```

Status

Inactive.

Implementation

```

* Check 1st CMBS Code
IF recm > 0
    SELECT tempcmbs
    SEEK temptr->xurnum
    IF FOUND()
        SELECT 0
        SELECT cmbscode
        SEEK tempcmbs->vcmbrcode
        IF .NOT. FOUND()
            STORE .T. TO errupd
            SELECT 4
            APPEND BLANK
            REPLACE errtext WITH urnum + "      Main CMBS Code Invalid " + tempcmbs-
>vcmbrcode
            REPLACE errnumber WITH "42"
            counter8 = counter8 +1
            SELECT 2
        ELSE
            * check code not excluded
            IF cmbscode->excluded = "Y"
                STORE .T. TO errupd
                SELECT 4
                APPEND BLANK
                REPLACE errtext WITH urnum + "      Main CMBS Code Excluded " + tempcmbs-
>vcmbrcode
                REPLACE errnumber WITH "43"
                counter8 = counter8 +1
                SELECT 2
            ENDIF
        ENDIF
    * add cmbs record order
    replace tempcmbs->vorder WITH 1
    SELECT tempcmbs
    skip
    lnorder = 2
    DO WHILE !EOF() .AND. tempcmbs->vurnum=temptr->xurnum
        replace tempcmbs->vorder WITH lnorder

```

```

        skip
        lnorder=lnorder+1
    enddo
    counter7 = counter7 +1
ELSE
    * CBMS code is maditory but initially this check will be disabled as some hospitals
will lodge incomplete records
    *STORE .T. TO errupd
    *SELECT 4
    *APPEND BLANK
    *REPLACE errtext WITH urnum +"      No CMBS Code Provided "
    *REPLACE errnumber WITH "44"
    SELECT 2
ENDIF
* delete old codes
SELECT cmbs
SEEK hosp+temptr->xurnum
IF !EOF ()
    DO WHILE cmbs->vpnumber=hosp+temptr->xurnum .AND. !EOF ()
        DELETE
        SKIP
        counter8 = counter8 +1
    ENDDO
ENDIF
SELECT 2
ENDIF
*
SELECT 2
skip
IF errupd
    STORE cterrt + 1 TO cterrt
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

45: Deferred Admission (Cat 4) With Treatment Status**45: Removal Reason Emergency (03) and Deferred Admisson (Cat 4) With Treatment Status****Pseudocode**

```

IF
(
    [Treatment Status] (data item 27) = 3 or 4
    AND
    [Admission Category] (data item 1) = 4
)
THEN
    PRINT error message 45: Deferred Admisson (Cat 4) With Treatment Status

(
    [Treatment Status] (data item 27) = 5
    AND
    [Admission Category] (data item 1) = 4
    AND
    [Reason For Removal] (data item 21) = 03
)
THEN
    PRINT error message 45: Removal Reason Emergency (03) and Deferred Admisson (Cat 4)
    With Treatment Status

```

Status

Active.

Implementation

```

*IF xstatus$'345' .and. xurgst='4' .and. xreasof='03'
IF (xstatus$'34' .and. xurgst='4') .or. (xstatus='5' .and. xurgst='4' .and. xreasof='03')
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    IF temptr->xstatus="5"
        REPLACE errtext WITH urnum +"      Removal Reason Emergency (03) and Deferred Admisson
(Cat 4) With Treatment Status - " + temptr->xstatus
    ELSE
        REPLACE errtext WITH urnum +"      Deferred Admisson (Cat 4) With Treatment Status - "
+ temptr->xstatus
    ENDIF
    REPLACE errnumber WITH "45"
    SELECT 2
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

46: Admit/Remove Date Prior TO Date Created**Pseudocode**

```

IF
(
    [Admission/Removal Date] (data item 2) < [Date Created]
    AND
    [Date Added To List] (data item 11) is not blank
)
THEN
    PRINT error message 46: Admit/Remove Date Prior TO Date Created

```

Status

Active.

Implementation

```

    *check for future date
IF xenddat > xcreatdate .and. dtoc(xenddat) <> " / / "
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH urnum + "    Admit/Remove Date prior TO Date Created - " +
dtoc(temptr->xenddat) + " < " + dtoc(temptr->xcreatdate)
    REPLACE errnumber WITH "46"
    SELECT 2
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

47: Excluded Legacy Indicator Procedure Code**Pseudocode**

```

IF
(
    [Intended Procedure] (data item 17) is valid
    AND
    (
        [Treatment Status] (data item 27) = 1 or 2
        OR
        [Admission/Removal Date] (data item 2) is blank
    )
)
THEN
    PRINT error message 47: Excluded Legacy Indicator Procedure Code

```

Status

Active.

Implementation

```

* check for invalid procedure code
STORE "J" + xcode TO gkey
SELECT 5
FIND &gkey
IF eof()
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH urnum + "      Invalid Procedure Code - " + temptr->xcode
    REPLACE errnumber WITH "15"
ELSE
    * note genend part of or section probably not needed
    IF (temptr->xdaton>genend .OR. (temptr->xstatus$'12' .AND. DATE()>=CTOD("01/08/2016")))
    .AND. dtoc(genend) <> " / / "
        * old code used
        STORE .T. TO errupd
        SELECT 4
        APPEND BLANK
        REPLACE errtext WITH urnum + "      Excluded Legacy Indicator Procedure Code - " +
temptr->xcode
        REPLACE errnumber WITH "47"
    ELSE
        SELECT 5
        IF temptr->xenddat>genend .AND. dtoc(genend) <> " / / " .AND. dtoc(temptr-
>xenddat) <> " / / "
            * old code used too late
            STORE .T. TO errupd
            SELECT 4
            APPEND BLANK
            REPLACE errtext WITH urnum + "      Excluded Legacy Indicator Procedure Code - " +
temptr->xcode
            REPLACE errnumber WITH "47"
    ENDIF
ENDIF

```

```

ENDIF
SELECT 5
IF temptr->xenddat<genstart .AND. dtoc(temptr->xenddat) <> " / / "
  * new code used too early
  STORE .T. TO errupd
  SELECT 4
  APPEND BLANK
  REPLACE errtext WITH urnum +"      Excluded Intended Procedure Code For Past Record - "
+ temptr->xcode
  REPLACE errnumber WITH "48"
ENDIF
ENDIF

```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

48: Excluded Intended Procedure Code For Past Record**Pseudocode**

```

IF
(
    [Intended Procedure] (data item 17) is valid
    AND
    [Admission/Removal Date] (data item 2) is blank
)
THEN
    PRINT error message 48: Excluded Intended Procedure Code For Past Record

```

Status

Active.

Implementation

```

* check for invalid procedure code
STORE "J" + xcode TO gkey
SELECT 5
FIND &gkey
IF eof()
    STORE .T. TO errupd
    SELECT 4
    APPEND BLANK
    REPLACE errtext WITH urnum + "          Invalid Procedure Code - " + temptr->xcode
    REPLACE errnumber WITH "15"
ELSE
    * note genend part of or section probably not needed
    IF (temptr->xdaton>genend .OR. (temptr->xstatus$'12' .AND. DATE())>=CTOD("01/08/2016")))
    .AND. dtoc(genend) <> " / / "
        * old code used
        STORE .T. TO errupd
        SELECT 4
        APPEND BLANK
        REPLACE errtext WITH urnum + "          Excluded Legacy Indicator Procedure Code - " +
temptr->xcode
        REPLACE errnumber WITH "47"
    ELSE
        SELECT 5
        IF temptr->xenddat>genend .AND. dtoc(genend) <> " / / " .AND. dtoc(temptr-
>xenddat) <> " / / "
            * old code used too late
            STORE .T. TO errupd
            SELECT 4
            APPEND BLANK
            REPLACE errtext WITH urnum + "          Excluded Legacy Indicator Procedure Code - " +
temptr->xcode
            REPLACE errnumber WITH "47"
        ENDIF
    ENDIF
    SELECT 5
    IF temptr->xenddat<genstart .AND. dtoc(temptr->xenddat) <> " / / "

```

```
* new code used too early
STORE .T. TO errupd
SELECT 4
APPEND BLANK
REPLACE errtext WITH urnum +"      Excluded Intended Procedure Code For Past Record - "
+ tempr->xcode
REPLACE errnumber WITH "48"
ENDIF
ENDIF
```

Issue

None.

Update data quality check

None.

Result

The pseudocode for this data quality check has been created.

Action

None.

.

For more information

**Data Governance and Integrity
Data, Analytics, and Insights
Australian Bragg Centre
North Terrace, Adelaide**
Email: DataAnalyticsandInsights@sa.gov.au
www.sahealth.sa.gov.au

For Official Use Only-I1-A1



www.ausgoal.gov.au/creative-commons



**Government
of South Australia**

SA Health