Australian Government

Australian Institute of Health and Welfare

Mental Health Establishments

Skeleton Data Set

2026-27 version 5.01

The metadata for SKL 5.01 can be found on the Online Validator metadata page

As at 27 February, 2026

694fe08ebe0862698c384b7eb4b243caac663db1

Table of Contents

1. Overview of the SKL

The data model for all mental health data sets includes a hierarchy describing the layers of mental health service delivery systems, for example, regional and organisational levels, and is an important design feature of the mental health collections. Ideally, the hierarchy reported by jurisdictions should be repeated among their various mental health data collections.

The Mental Health Information Strategy Standing Committee (MHISSC), and its National Minimum Data Set Subcommittee, agreed to a number of key principles to scrutinise the hierarchies reported in the various mental health data sets. In essence, the committees agreed to provide a 'Skeleton' of the Mental Health Establishments (MHE) National Minimum Dataset (NMDS), to be known as an SKL dataset, at the same time in the reporting cycle as the Community Mental Health Care and Residential Mental Health Care NMDSs, and the National Outcomes and Casemix Collection (NOCC). This 'Skeleton' is considered the 'gold standard' against which all other mental health data files are compared, including the CMHC, RMHC, NOCC and MHE files.

The Online Validator supports the mental health collections as the vehicle for checking data submissions and provides the necessary interactive space for discussions about data issues between various stakeholders. The Online Validator has the capacity to check the hierarchy between collections, as well as identifying changes compared to previous submissions, using the comparisons with the SKL file.

The purpose of this module is to outline the layout and format of the Mental Health Establishments Skeleton (SKL) dataset to be submitted by States and Territories to the Australian Institute of Health and Welfare (AIHW) and Department of Health, Disability and Ageing in respect of the 2026-27 year. The file is identical in structure to an MHE file, however, is limited to key components of the MHE file in order to facilitate the 'between-data set' comparisons.

1.1. Changes for 2026-27

The specific detailed changes to the 2026-27 (version 5.01) specifications, compared to 2025-26 (version 5.00) are listed below.

1.1.1. Changes to the data model

No changes were made to the data model.

1.1.2. Changes to definitions

No changes were made to the definitions.

1.2. Principles and agreements

The SKL submission has been built using the following principles.

Table 1.1 SKL principles

	Decision	Notes
Timeline	December of the reporting cycle.	The supply of the MHE NMDS skeleton should occur in December of the reporting cycle, along with CMHC, RMHC and NOCC files, noting that it would be ideal to be able to supply the file earlier. The MHE Skeleton file does not have to be submitted before all other file types for the collection year. The processes comparing between files will only occur once a relevant file has been loaded. However, between file comparisons will only be made on reviewer accepted files, i.e. those that have passed stage 1 validation. When a replacement SKL file is submitted, all dependant files will be re-compared with the revised SKL automatically.
File type	MHE Skeleton file is an independent file type	This assists the processes managed by the Online Validator. When uploading a file, there will be an option to specify if the file is a MHE Skeleton submission or a full MHE file.
File structure	Structurally equivalent to the full MHE file.	The MHE skeleton file must meet Stage 1 - structural Compliance tests for any MHE file, for example, line lengths and zero filling. An appropriate set of Stage 1 - Structural Checks will be undertaken on the mandatory data elements of the MHE NMDS skeleton file, including checks for malformed, duplicate, orphan, barren and miscoded records, plus missing data in mandatory fields. Importantly, failure of these rules will require resubmission of the MHE NMDS skeleton file, in accordance with existing file submission processes.
File content	The mandatory items included in the MHE NMDS skeleton include only those identifiers and attributes that permit comparison between the NMDSs.	All non-mandatory items will be blanked out with spaced during the file submission process. Note that this does not prevent jurisdictions from providing 'real' data in non-mandatory data elements when loading the file, however, blanking out this information will ensure confidentiality is maintained as part of the upload process.
Historical comparisons	An appropriate set of Stage 2—Historical Checks will be made on the entities in the MHE NMDS skeleton file, focusing on attributes that will be used for comparisons with the other data sets.	This will generate an issues list that will require jurisdictional input as per existing validation processes. Logicly is investigating the technical feasibility of automatically transferring any responses made to the issues lists at this stage across to the replacement MHE file, as per existing processes, when the 'complete MHE NMDS' file is proposed.
Parental exclusions	Issues raised at each subsequent level of analysis, starting at Regional level and progressing to Service Unit level, excludes issues related to non-matching parent entities.	This will reduce the burden on jurisdictions to respond to issues generated by problems with the parent entity. By way of example, if a region in the MHE skeleton cannot be identified in a RMHC file, issues would not be generated for any subsequent Organisations and Service Units within that region.
Attributes not considered for comparison	Entity names Entity geography	At this stage, the comparison of these attributes is not considered a high priority. This decision may be revisited in the future.
Comparison outputs	Issues list and reports.	Each non-matching entity will generate an issue in a similar way to existing issue list generation. A summary 'Validation MHE Comparison Report' will be developed to supplement the existing reports for each of the MHE, CMHC, RMHC and NOCC report views.

1.3. Comparisons

The following comparisons between the SKL and other data sets are made in the Online Validator. Comparisons between data sets are made only on those entities that are logical to compare. For example, an organisation in the SKL may only have admitted and ambulatory services. In this case, a check for the existence of a service unit between the SKL and the RMHC file would not be logical.

1.3.1. Region and organisation level comparisons

Regional and organisation entity matching is considered a minimum requirement, regardless of the collections being compared.

1.3.2. SKL vs. RMHC service unit level comparisons

SKL vs. RMHC comparisons will be made according to the following.

Analysis level	SKL data element	RMHC data element
Region level	RegId	RegId
Organisational level	RegId, OrgId	RegId, OrgId
Service unit	RegId, OrgId, ResiId	RegId, OrgId, SUId

1.3.3. SKL vs. CMHC service unit level comparisons

SKL vs. CMHC comparisons will be made according to the following.

	SKL data element	CMHC data element
Region level	RegId	RegId
Organisational level	RegId, OrgId	RegId, OrgId
Service unit level	RegId, OrgId, AMBU_TargetPop	RegId, OrgId, SERV.TargetPop

1.3.4. SKL vs. NOCC entity comparisons

SKL vs. NOCC comparisons will be made according to the following.

Analysis level	SKL data element	NOCC data element
Region level	RegId	RegId
Organisational level	RegId, OrgId	RegId, OrgId
Hospital level	RegId, OrgId, HospId	RegId, OrgId, HospId
Admitted service unit	RegId, OrgId, AdmiId	RegId, OrgId, SUId, SUType=1

Analysis level	SKL data element	NOCC data element
Residential service unit	RegId, OrgId, ResiId	RegId, OrgId, SUId, SUType=2
Ambulatory service unit	RegId, OrgId, AMBU_TargetPop	RegId, OrgId, (SUId_Targetpop,SUType=3)

1.3.5. Data Integrity

For cases of missing data (that is, unknown, not stated or not available):

- For **Numeric [Num] fields**, the data should be reported as zero, using leading zeros when necessary to pad out the field to the required length. The principle here is that all numeric fields require a valid value.
- For **Text [Char] fields**, the data should be space-filled to the required length. For single character fields where a 'missing/not stated' value has been specified for a particular data element (for example, '9' has been specified for missing data), use the stated value for 'missing/not stated' rather than simply space filling.

Values in **Date** [**Date**] fields must be recorded in compliance with the standard format used across the *National health data dictionary*; specifically, dates must be of fixed 8 column width in the format DDMMYYYY, with leading zeros used when necessary to pad out a value. For instance, 13 March 2027 would appear as 13032027.

Values in **Numeric** [**Num**] fields must be zero-filled and right-justified. These should consist only of the numerals 0 to 9 and the decimal ('.') point if applicable to the data element.

Note: Fields defined as 'Numeric' are those that have numeric properties—that is, the values, for example, can be added or subtracted in a manner that is valid. Where a field uses numeric characters that do not have these properties (for example, the use of numbers for *Patient identifier*), the field is defined as 'Character'.

Values in **Character** [**Char**] fields must be left justified and space-filled. These should consist of any of the printable ASCII character set (that is, excluding control codes such as newline, bell and linefeed).

1.4. Dataset specifications (DSS)

The following tables specify the order in which the data items should be provided to the AIHW.

The extract format consists of a set of hierarchically ordered *Data records*, of which there are fourteen types (see Table 1.2). In each extract file for any given period, the *Data records* must be preceded by a single *File Header Record* having the structure outlined in Table 1.3.

All records presented in the extract file should be grouped in the following order: Header Record; State/Territory details records; State MH NGO details records; Region details records; Region MH NGO details records; Organisation details records; Organisation full-time equivalent staff by service setting details records; Hospital/ Service unit cluster details records; and Service unit details records.

With the exception of State MH NGO, Region, Region MH NGO, Organisation and Service unit cluster details records, all data records should include the following elements in the order shown:

- Record type
- Establishment identifier (comprising: *State/Territory identifier*; *Region identifier*; *Organisation identifier*; *Hospital identifier/Service unit cluster identifier*; and *Service unit identifier*)
- Specific data in the format specified for the given record type.

The State MH NGOE and Region MH NGOE payments records use different Establishment identifier compositions:

- State MH NGOE: *State/Territory identifier*; and *Mental health non-government organisation identifier*
- Region MH NGOE: *State/Territory identifier*; *Region identifier*; and *Mental health non-government organisation identifier*

The order of fields in a record must be the same as the order they are listed in the Record Layouts specified below. Field values should be formatted as specified in the Record Layouts.

The first field in each record must be *Record Type*. Valid values for *Record Type* are shown in Table 1.2.

Table 1.2 Valid values for Record Type

Record Type	Description
HR	File Header Record
ST	State/Territory details
STNGO	State-level non-government organisation details
STNGOE	State MH NGOE payments
REG	Region details
REGNGO	Region-level non-government organisation details
REGNGOE	Region MH NGOE payments
ORG	Organisation details
FTEORG	Organisation full-time equivalent staff by service setting details
HOSP	Hospital details
CLUS	Service unit cluster details
ADMI	Admitted patient service unit details
AMBU	Ambulatory service unit details
RESI	Residential service unit details

NGOE records may be supplied but do not contribute to the SKL checking processes.

1.4.1. File header record

The first record of the extract file must be a File Header Record (*Record Type* = 'HR'), and it must be the only such record in the file.

The File Header Record is a quality control mechanism, which uniquely identifies each file that is submitted to the Online Validator.

The layout of the File Header Record is shown in Table 1.3.

Table 1.3 Record Layout for File Header Record within the data extract Data record layout

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Record Type `RecType`	Char[8]	1	—	Value = *HR*
State/Territory Identifier [1] `State`	Char[1]	9	790405	**1:** New South Wales **2:** Victoria **3:** Queensland **4:** South Australia **5:** Western Australia **6:** Tasmania **7:** Northern Territory **8:** Australian Capital Territory
Batch Number `BatchNo`	Char[9]	10	—	Represents the YYYYNNNNN component of the extract file name.
Report Period Start Date `RepStart`	Date[8]	19	—	Report period start date
Report Period End Date `RepEnd`	Date[8]	27	—	Report period end date
Data File Generation Date `GenDt`	Date[8]	35	—	Data file generation date
Data File Type `FileType`	Char[3]	43	—	Value = *SKL*

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
SKL Specification Version Number `SpecVer`	Char[5]	46	—	Value = *05.01*

Record length = 50

Notes

[1]
(METEOR includes code 9, but that is not applicable to SKL)

1.4.2. State/Territory data record

The extract format for the *Data records* is specified in detail in Table 1.4 to Table 1.16. The order of fields in each record must be the same as the order they are shown below. Field values should be formatted as specified.

Table 1.4 Data record layout - State/Territory details

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Record Type `RecType`	Char[8]	1	—	Value = *ST*
State/Territory Identifier [2] `State`	Char[1]	9	790405	**1:** New South Wales **2:** Victoria **3:** Queensland **4:** South Australia **5:** Western Australia **6:** Tasmania **7:** Northern Territory **8:** Australian Capital Territory
State/Territory Name `StateName`	Char[28]	10	—	Name used to identify the State/Territory
Blank 186 Spacing Field `Blank186`	Blank[186]	38	—	Field value is ignored for SKL processing, can contain spaces or MHE values

Record length = 223

Notes

[2]
(METEOR includes code 9, but that is not applicable to SKL)

1.4.3. State MH NGO details record

Table 1.5 Data record layout - State MH NGO details

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Record Type `RecType`	Char[8]	1	—	Value = *STNGO*
State/ Territory Identifier [3] `State`	Char[1]	9	790405	**1:** New South Wales **2:** Victoria **3:** Queensland **4:** South Australia **5:** Western Australia **6:** Tasmania **7:** Northern Territory **8:** Australian Capital Territory
Non-Government Organisation Identifier `NGOId`	Char[11]	10	795673	A unique identifier for Non-Government Organisations (NGO) established by jurisdictions. Where the NGO is a Primary Health Network, the permissible identifiers are in accordance to the Guide for use for this data element.
Non-Government Organisation Name `NGOName`	Char[100]	21	795695	Name used to identify the Non-Government Organisation (NGO). Where the NGO is a Primary Health Network, the permissible identifiers are in accordance to the Guide for use for this data element.

Record length = 120

Notes

[3]
(METEOR includes code 9, but that is not applicable to SKL)

1.4.4. State MH NGOE Payments data record

Table 1.6 Data record layout - State MH NGOE Payments

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Record Type `RecType`	Char[8]	1	—	Value = *STNGOE*
State/Territory Identifier [4] `State`	Char[1]	9	790405	**1:** New South Wales **2:** Victoria **3:** Queensland **4:** South Australia **5:** Western Australia **6:** Tasmania **7:** Northern Territory **8:** Australian Capital Territory
Non-Government Organisation Identifier `NGOId`	Char[11]	10	795673	A unique identifier for Non-Government Organisations (NGO) established by jurisdictions. Where the NGO is a Primary Health Network, the permissible identifiers are in accordance to the Guide for use for this data element.
Blank 11 Spacing Field `Blank11`	Blank[11]	21	—	Field value is ignored for SKL processing, can contain spaces or MHE values

Record length = 31

Notes

[4]
(METEOR includes code 9, but that is not applicable to SKL)

1.4.5. Region data record

Table 1.7 Data record layout - Region details

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Record Type `RecType`	Char[8]	1	—	Value = *REG*

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
State/ Territory Identifier [5] `State`	Char[1]	9	790405	**1:** New South Wales **2:** Victoria **3:** Queensland **4:** South Australia **5:** Western Australia **6:** Tasmania **7:** Northern Territory **8:** Australian Capital Territory
Region Identifier `RegId`	Char[2]	10	269940	AA: Region (values as specified by individual jurisdiction). Identifiers used in this collection should map to the identifiers used in data for the NMDSs for Community Mental Health Care and Residential Mental Health Care.
Region Name `RegName`	Char[60]	12	407187	Common name used to identify the Region.
Blank 180 Spacing Field `Blank180`	Blank[180]	72	—	Field value is ignored for SKL processing, can contain spaces or MHE values

Record length = 251

Notes

[5]
(METEOR includes code 9, but that is not applicable to SKL)

1.4.6. Region MH NGO details record

Table 1.8 Data record layout - Region MH NGO details

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Record Type `RecType`	Char[8]	1	—	Value = *REGNGO*

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
State/ Territory Identifier [6] `State`	Char[1]	9	790405	**1:** New South Wales **2:** Victoria **3:** Queensland **4:** South Australia **5:** Western Australia **6:** Tasmania **7:** Northern Territory **8:** Australian Capital Territory
Region Identifier `RegId`	Char[2]	10	269940	AA: Region (values as specified by individual jurisdiction). Identifiers used in this collection should map to the identifiers used in data for the NMDSs for Community Mental Health Care and Residential Mental Health Care.
Non-Government Organisation Identifier `NGOId`	Char[11]	12	795673	A unique identifier for Non-Government Organisations (NGO) established by jurisdictions. Where the NGO is a Primary Health Network, the permissible identifiers are in accordance to the Guide for use for this data element.
Non-Government Organisation Name `NGOName`	Char[100]	23	795695	Name used to identify the Non-Government Organisation (NGO). Where the NGO is a Primary Health Network, the permissible identifiers are in accordance to the Guide for use for this data element.

Record length = 122

Notes

[6]
(METEOR includes code 9, but that is not applicable to SKL)

1.4.7. Region MH NGOE Payments data record

Table 1.9 Data record layout - Region MH NGOE Payments

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Record Type `RecType`	Char[8]	1	—	Value = *REGNGOE*

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
State/ Territory Identifier [7] `State`	Char[1]	9	790405	**1:** New South Wales **2:** Victoria **3:** Queensland **4:** South Australia **5:** Western Australia **6:** Tasmania **7:** Northern Territory **8:** Australian Capital Territory
Region Identifier `RegId`	Char[2]	10	269940	AA: Region (values as specified by individual jurisdiction). Identifiers used in this collection should map to the identifiers used in data for the NMDSs for Community Mental Health Care and Residential Mental Health Care.
Non-Government Organisation Identifier `NGOId`	Char[11]	12	795673	A unique identifier for Non-Government Organisations (NGO) established by jurisdictions. Where the NGO is a Primary Health Network, the permissible identifiers are in accordance to the Guide for use for this data element.
Blank 11 Spacing Field `Blank11`	Blank[11]	23	—	Field value is ignored for SKL processing, can contain spaces or MHE values

Record length = 33

Notes

[7]
(METEOR includes code 9, but that is not applicable to SKL)

1.4.8. Organisation data record

Table 1.10 Data record layout - Organisation Details

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Record Type `RecType`	Char[8]	1	—	Value = *ORG*

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
State/ Territory Identifier [8] `State`	Char[1]	9	790405	**1:** New South Wales **2:** Victoria **3:** Queensland **4:** South Australia **5:** Western Australia **6:** Tasmania **7:** Northern Territory **8:** Australian Capital Territory
Region Identifier `RegId`	Char[2]	10	269940	AA: Region (values as specified by individual jurisdiction). Identifiers used in this collection should map to the identifiers used in data for the NMDSs for Community Mental Health Care and Residential Mental Health Care.
Organisation Identifier `OrgId`	Char[9]	12	795837	A(9): Mental health service organisation identifier. Identifiers used in this collection should map to the identifiers used in data for the NMDSs for Community Mental Health Care and Residential Mental Health Care.
Organisation Name `OrgName`	Char[100]	21	405767	Common name used to identify the Organisation
Blank 539 Spacing Field `Blank539`	Blank[539]	121	—	Field value is ignored for SKL processing, can contain spaces or MHE values

Record length = 659

Notes

[8]
(METEOR includes code 9, but that is not applicable to SKL)

1.4.9. Organisation: FTE staff by Service Setting data record

Table 1.11 Data record layout - Organisation: FTE staff by Service Setting data record

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Record Type `RecType`	Char[8]	1	—	Value = *FTEORG*

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
State/ Territory Identifier [9] `State`	Char[1]	9	790405	**1:** New South Wales **2:** Victoria **3:** Queensland **4:** South Australia **5:** Western Australia **6:** Tasmania **7:** Northern Territory **8:** Australian Capital Territory
Region Identifier `RegId`	Char[2]	10	269940	AA: Region (values as specified by individual jurisdiction). Identifiers used in this collection should map to the identifiers used in data for the NMDSs for Community Mental Health Care and Residential Mental Health Care.
Organisation Identifier `OrgId`	Char[9]	12	795837	A(9): Mental health service organisation identifier. Identifiers used in this collection should map to the identifiers used in data for the NMDSs for Community Mental Health Care and Residential Mental Health Care.
Service Setting `Setting`	Char[1]	21	493347	**1:** Admitted patient care setting **2:** Residential care setting **3:** Ambulatory care setting **4:** Organisational overhead setting
Blank 57 Spacing Field `Blank57`	Blank[57]	22	—	Field value is ignored for SKL processing, can contain spaces or MHE values

Record length = 78

Notes

[9]
(METEOR includes code 9, but that is not applicable to SKL)

1.4.10. Hospital data record

Table 1.12 Data record layout - Hospital details

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Record Type `RecType`	Char[8]	1	—	Value = *HOSP*
State/ Territory Identifier [10] `State`	Char[1]	9	790405	**1:** New South Wales **2:** Victoria **3:** Queensland **4:** South Australia **5:** Western Australia **6:** Tasmania **7:** Northern Territory **8:** Australian Capital Territory
Region Identifier `RegId`	Char[2]	10	269940	AA: Region (values as specified by individual jurisdiction). Identifiers used in this collection should map to the identifiers used in data for the NMDSs for Community Mental Health Care and Residential Mental Health Care.
Organisation Identifier `OrgId`	Char[9]	12	795837	A(9): Mental health service organisation identifier. Identifiers used in this collection should map to the identifiers used in data for the NMDSs for Community Mental Health Care and Residential Mental Health Care.
Hospital Identifier `HospId`	Char[9]	21	795844	A(9): Hospital identifier (equals Establishment number as reported for NMDS for Admitted Patient Care)
Sector `Sector`	Char[1]	30	269977	**1:** Public **2:** Private
Blank 1 Spacing Field `Blank1`	Blank[1]	31	—	Field value is ignored for SKL processing, can contain spaces or MHE values
Hospital Name `HospName`	Char[100]	32	407430	Common name used to identify the hospital.
Blank 10 Spacing Field `Blank10`	Blank[10]	132	—	Field value is ignored for SKL processing, can contain spaces or MHE values

Record length = 141

Notes

[10]
(METEOR includes code 9, but that is not applicable to SKL)

1.4.11. Service Unit Cluster data record

Table 1.13 Data record layout - Service Unit Cluster Details

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Record Type `RecType`	Char[8]	1	—	Value = *CLUS*
State/ Territory Identifier [11] `State`	Char[1]	9	790405	**1:** New South Wales **2:** Victoria **3:** Queensland **4:** South Australia **5:** Western Australia **6:** Tasmania **7:** Northern Territory **8:** Australian Capital Territory
Region Identifier `RegId`	Char[2]	10	269940	AA: Region (values as specified by individual jurisdiction). Identifiers used in this collection should map to the identifiers used in data for the NMDSs for Community Mental Health Care and Residential Mental Health Care.
Organisation Identifier `OrgId`	Char[9]	12	795837	A(9): Mental health service organisation identifier. Identifiers used in this collection should map to the identifiers used in data for the NMDSs for Community Mental Health Care and Residential Mental Health Care.
Service Unit Cluster Identifier `ClusId`	Char[9]	21	795848	A(9): An identifier to indicate that a service unit is one of a cluster of service units, defined through administrative or clinical governance arrangements. If no cluster applies, set to 00000. As this field enables linking with the NMDSs for Community Mental Health Care and Residential Mental Health Care, the identifiers used in this collection should be the same.
Service Unit Cluster Name `ClusName`	Char[100]	30	409209	If no cluster applies, enter organisation name as appears in previous line.

Record length = 129

Notes

[11]
(METEOR includes code 9, but that is not applicable to SKL)

1.4.12. Admitted Patient Service Unit data record

Table 1.14 Data record layout - Admitted Patient Service Unit Details

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Record Type `RecType`	Char[8]	1	—	Value = *ADMI*
State/ Territory Identifier [12] `State`	Char[1]	9	790405	**1:** New South Wales **2:** Victoria **3:** Queensland **4:** South Australia **5:** Western Australia **6:** Tasmania **7:** Northern Territory **8:** Australian Capital Territory
Region Identifier `RegId`	Char[2]	10	269940	AA: Region (values as specified by individual jurisdiction). Identifiers used in this collection should map to the identifiers used in data for the NMDSs for Community Mental Health Care and Residential Mental Health Care.
Organisation Identifier `OrgId`	Char[9]	12	795837	A(9): Mental health service organisation identifier. Identifiers used in this collection should map to the identifiers used in data for the NMDSs for Community Mental Health Care and Residential Mental Health Care.
Hospital Identifier `HospId`	Char[9]	21	795844	A(9): Hospital identifier (equals Establishment number as reported for NMDS for Admitted Patient Care)
Admitted Patient Service Unit Identifier `AdmiId`	Char[9]	30	795850	A(9): Service unit identifier. Identifiers used in this collection should map to the identifiers used in data for the NMDSs for Community Mental Health Care and Residential Mental Health Care.
Blank 3 Spacing Field `Blank3`	Blank[3]	39	—	Field value is ignored for SKL processing, can contain spaces or MHE values

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Target Population [13] `TargetPop`	Char[1]	42	682403	**1:** Child and adolescent **2:** Older person **3:** Forensic **4:** General **5:** Youth **7:** Not applicable
Blank 1 Spacing Field `Blank1`	Blank[1]	43	—	Field value is ignored for SKL processing, can contain spaces or MHE values
Admitted Patient Service Unit Name `AdmiName`	Char[100]	44	721830	Common name used to identify the service unit.
Blank 58 Spacing Field `Blank58`	Blank[58]	144	—	Field value is ignored for SKL processing, can contain spaces or MHE values

Record length = 201

Notes

[12]
(METEOR includes code 9, but that is not applicable to SKL)

[13]
Code 7 only applies to FTEORG usage, METEOR code 9 does not apply

1.4.13. Ambulatory Service Unit data record

Table 1.15 Data record layout - Ambulatory Service Unit Details

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Record Type `RecType`	Char[8]	1	—	Value = *AMBU*

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
State/ Territory Identifier [14] `State`	Char[1]	9	790405	**1:** New South Wales **2:** Victoria **3:** Queensland **4:** South Australia **5:** Western Australia **6:** Tasmania **7:** Northern Territory **8:** Australian Capital Territory
Region Identifier `RegId`	Char[2]	10	269940	AA: Region (values as specified by individual jurisdiction). Identifiers used in this collection should map to the identifiers used in data for the NMDSs for Community Mental Health Care and Residential Mental Health Care.
Organisation Identifier `OrgId`	Char[9]	12	795837	A(9): Mental health service organisation identifier. Identifiers used in this collection should map to the identifiers used in data for the NMDSs for Community Mental Health Care and Residential Mental Health Care.
Service Unit Cluster Identifier `ClusId`	Char[9]	21	795848	A(9): An identifier to indicate that a service unit is one of a cluster of service units, defined through administrative or clinical governance arrangements. If no cluster applies, set to 00000. As this field enables linking with the NMDSs for Community Mental Health Care and Residential Mental Health Care, the identifiers used in this collection should be the same.
Ambulatory Service Unit Identifier `AmbuId`	Char[9]	30	795855	A(9): Service unit identifier. Identifiers used in this collection should map to the identifiers used in data for the NMDSs for Community Mental Health Care and Residential Mental Health Care.
Blank 3 Spacing Field `Blank3`	Blank[3]	39	—	Field value is ignored for SKL processing, can contain spaces or MHE values
Target Population [15] `TargetPop`	Char[1]	42	682403	**1:** Child and adolescent **2:** Older person **3:** Forensic **4:** General **5:** Youth **7:** Not applicable

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Sector `Sector`	Char[1]	43	269977	**1:** Public **2:** Private
Blank 1 Spacing Field `Blank1`	Blank[1]	44	—	Field value is ignored for SKL processing, can contain spaces or MHE values
Ambulatory Service Unit Name `AmbuName`	Char[100]	45	750374	Common name used to identify the service unit.
Blank 49 Spacing Field `Blank49`	Blank[49]	145	—	Field value is ignored for SKL processing, can contain spaces or MHE values

Record length = 193

Notes

[14]
(METEOR includes code 9, but that is not applicable to SKL)

[15]
Code 7 only applies to FTEORG usage, METEOR code 9 does not apply

1.4.14. Residential Service Unit data record

Table 1.16 Data record layout - Residential Service Unit Details

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Record Type `RecType`	Char[8]	1	—	Value = *RESI*

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
State/ Territory Identifier [16] `State`	Char[1]	9	790405	**1:** New South Wales **2:** Victoria **3:** Queensland **4:** South Australia **5:** Western Australia **6:** Tasmania **7:** Northern Territory **8:** Australian Capital Territory
Region Identifier `RegId`	Char[2]	10	269940	AA: Region (values as specified by individual jurisdiction). Identifiers used in this collection should map to the identifiers used in data for the NMDSs for Community Mental Health Care and Residential Mental Health Care.
Organisation Identifier `OrgId`	Char[9]	12	795837	A(9): Mental health service organisation identifier. Identifiers used in this collection should map to the identifiers used in data for the NMDSs for Community Mental Health Care and Residential Mental Health Care.
Service Unit Cluster Identifier `ClusId`	Char[9]	21	795848	A(9): An identifier to indicate that a service unit is one of a cluster of service units, defined through administrative or clinical governance arrangements. If no cluster applies, set to 00000. As this field enables linking with the NMDSs for Community Mental Health Care and Residential Mental Health Care, the identifiers used in this collection should be the same.
Residential Service Unit Identifier `ResiId`	Char[9]	30	795859	A(9): Service unit identifier. Identifiers used in this collection should map to the identifiers used in data for the NMDSs for Community Mental Health Care and Residential Mental Health Care.
Blank 3 Spacing Field `Blank3`	Blank[3]	39	—	Field value is ignored for SKL processing, can contain spaces or MHE values
Target Population [17] `TargetPop`	Char[1]	42	682403	**1:** Child and adolescent **2:** Older person **3:** Forensic **4:** General **5:** Youth **7:** Not applicable

Data Element	Type [Length]	Start	METEOR Identifier	Notes / Values
Blank 2 Spacing Field `Blank2`	Blank[2]	43	—	Field value is ignored for SKL processing, can contain spaces or MHE values
Sector `Sector`	Char[1]	45	269977	**1:** Public **2:** Private
Blank 1 Spacing Field `Blank1`	Blank[1]	46	—	Field value is ignored for SKL processing, can contain spaces or MHE values
Residential Service Unit Name `ResiName`	Char[100]	47	722715	Common name used to identify the service unit.
Blank 55 Spacing Field `Blank55`	Blank[55]	147	—	Field value is ignored for SKL processing, can contain spaces or MHE values

Record length = 201

Notes

[16]
(METEOR includes code 9, but that is not applicable to SKL)

[17]
Code 7 only applies to FTEORG usage, METEOR code 9 does not apply

1.5. Data elements

1.5.1. Admitted Patient Service Unit Identifier

Domain:
A(9): Service unit identifier. Identifiers used in this collection should map to the identifiers used in data for the NMDSs for Community Mental Health Care and Residential Mental Health Care.
Field Name:
`AdmiId`
METEOR Identifier:
795850

1.5.2. Admitted Patient Service Unit Name

Domain:
Common name used to identify the service unit.

Field Name:

`AdmiName`

METEOR Identifier:

721830

1.5.3. Ambulatory Service Unit Identifier

Domain:

A(9): Service unit identifier. Identifiers used in this collection should map to the identifiers used in data for the NMDSs for Community Mental Health Care and Residential Mental Health Care.

Field Name:

`AmbuId`

METEOR Identifier:

795855

1.5.4. Ambulatory Service Unit Name

Domain:

Common name used to identify the service unit.

Field Name:

`AmbuName`

METEOR Identifier:

750374

1.5.5. Batch Number

Definition:

Represents the YYYYNNNNN component of the extract file name.

Field Name:

`BatchNo`

1.5.6. Blank 1 Spacing Field

Definition:

Field value is ignored for SKL processing, can contain spaces or MHE values

Field Name:

`Blank1`

1.5.7. Blank 2 Spacing Field

Definition:

Field value is ignored for SKL processing, can contain spaces or MHE values

Field Name:

`Blank2`

1.5.8. Blank 3 Spacing Field

Definition:
Field value is ignored for SKL processing, can contain spaces or MHE values
Field Name:
`Blank3`

1.5.9. Blank 10 Spacing Field

Definition:
Field value is ignored for SKL processing, can contain spaces or MHE values
Field Name:
`Blank10`

1.5.10. Blank 11 Spacing Field

Definition:
Field value is ignored for SKL processing, can contain spaces or MHE values
Field Name:
`Blank11`

1.5.11. Blank 49 Spacing Field

Definition:
Field value is ignored for SKL processing, can contain spaces or MHE values
Field Name:
`Blank49`

1.5.12. Blank 54 Spacing Field

Definition:
Field value is ignored for SKL processing, can contain spaces or MHE values
Field Name:
`Blank54`

1.5.13. Blank 55 Spacing Field

Definition:
Field value is ignored for SKL processing, can contain spaces or MHE values
Field Name:
`Blank55`

1.5.14. Blank 57 Spacing Field

Definition:
Field value is ignored for SKL processing, can contain spaces or MHE values
Field Name:
`Blank57`

1.5.15. Blank 58 Spacing Field

Definition:
Field value is ignored for SKL processing, can contain spaces or MHE values
Field Name:
`Blank58`

1.5.16. Blank 180 Spacing Field

Definition:
Field value is ignored for SKL processing, can contain spaces or MHE values
Field Name:
`Blank180`

1.5.17. Blank 186 Spacing Field

Definition:
Field value is ignored for SKL processing, can contain spaces or MHE values
Field Name:
`Blank186`

1.5.18. Blank 189 Spacing Field

Definition:
Field value is ignored for SKL processing, can contain spaces or MHE values
Field Name:
`Blank189`

1.5.19. Blank 195 Spacing Field

Definition:
Field value is ignored for SKL processing, can contain spaces or MHE values
Field Name:
`Blank195`

1.5.20. Blank 539 Spacing Field

Definition:
Field value is ignored for SKL processing, can contain spaces or MHE values
Field Name:
`Blank539`

1.5.21. Blank 548 Spacing Field

Definition:
Field value is ignored for SKL processing, can contain spaces or MHE values
Field Name:
`Blank548`

1.5.22. Data File Generation Date

Definition:
Data file generation date
Domain:
Data file generation date

Valid date expressed as DDMMYYYY
Field Name:
`GenDt`

1.5.23. Data File Type

Definition:
Data file type
Domain:
Value = *SKL*
Field Name:
`FileType`

1.5.24. Hospital Identifier

Domain:
A(9): Hospital identifier (equals Establishment number as reported for NMDS for Admitted Patient Care)
Field Name:
`HospId`
METEOR Identifier:
795844

1.5.25. Hospital Name

Domain:
Common name used to identify the hospital.
Field Name:
`HospName`
METEOR Identifier:
407430

1.5.26. Non-Government Organisation Identifier

Domain:
A unique identifier for Non-Government Organisations (NGO) established by jurisdictions.

Where the NGO is a Primary Health Network, the permissible identifiers are in accordance to the Guide for use for this data element.
Field Name:
`NGOId`
METEOR Identifier:
795673

1.5.27. Non-Government Organisation Name

Domain:

Name used to identify the Non-Government Organisation (NGO). Where the NGO is a Primary Health Network, the permissible identifiers are in accordance to the Guide for use for this data element.

Field Name:

`NGOName`

METEOR Identifier:

795695

1.5.28. Organisation Identifier

Definition:

Mental health service organisation identifier.

Domain:

A(9): Mental health service organisation identifier.

Identifiers used in this collection should map to the identifiers used in data for the NMDSs for Community Mental Health Care and Residential Mental Health Care.

Field Name:

`OrgId`

METEOR Identifier:

795837

1.5.29. Organisation Name

Definition:

Common name used to identify the Organisation

Field Name:

`OrgName`

METEOR Identifier:

405767

1.5.30. Record Type

Definition:

A code indicating the type of each record included in a SKL data file.

Domain:

ADMI:

Admitted patient service unit details

AMBU:

Ambulatory service unit details

CLUS:

Service unit cluster details

FTEORG:

Organisation full-time equivalent staff by service setting details

HOSP:

Hospital details

HR:

File header record

ORG:

Organisation details

REG:

Region details

REGNGO:

Region-level non-government organisation details

REGNGOE:

Region-level non-government organisation payments

RESI:

Residential service unit details

ST:

State/Territory details

STNGO:

State-level non-government organisation details

STNGOE:

State-level non-government organisation payments

Field Name:

`RecType`

1.5.31. Region Identifier

Domain:

AA: Region (values as specified by individual jurisdiction). Identifiers used in this collection should map to the identifiers used in data for the NMDSs for Community Mental Health Care and Residential Mental Health Care.

Field Name:

`RegId`

METEOR Identifier:

269940

1.5.32. Region Name

Definition:

Common name used to identify the Region.

Field Name:

`RegName`

METEOR Identifier:

407187

1.5.33. Report Period End Date

Definition:
Report period end date
Domain:
Report period end date

Valid date expressed as DDMMYYYY
Field Name:
`RepEnd`

1.5.34. Report Period Start Date

Definition:
Report period start date
Domain:
Report period start date

Valid date expressed as DDMMYYYY
Field Name:
`RepStart`

1.5.35. Residential Service Unit Identifier

Domain:
A(9): Service unit identifier. Identifiers used in this collection should map to the identifiers used in data for the NMDSs for Community Mental Health Care and Residential Mental Health Care.
Field Name:
`ResiId`
METEOR Identifier:
795859

1.5.36. Residential Service Unit Name

Domain:
Common name used to identify the service unit.
Field Name:
`ResiName`
METEOR Identifier:
722715

1.5.37. Sector

Domain:
1:
Public
2:
Private

Field Name:

`Sector`

METEOR Identifier:

269977

1.5.38. Service Setting

Domain:

1:

Admitted patient care setting

2:

Residential care setting

3:

Ambulatory care setting

4:

Organisational overhead setting

Field Name:

`Setting`

METEOR Identifier:

493347

1.5.39. Service Unit Cluster Identifier

Domain:

A(9): An identifier to indicate that a service unit is one of a cluster of service units, defined through administrative or clinical governance arrangements. If no cluster applies, set to 00000. As this field enables linking with the NMDSs for Community Mental Health Care and Residential Mental Health Care, the identifiers used in this collection should be the same.

Field Name:

`ClusId`

METEOR Identifier:

795848

1.5.40. Service Unit Cluster Name

Definition:

Common name used to identify the service unit cluster.

Domain:

If no cluster applies, enter organisation name as appears in previous line.

Field Name:

`ClusName`

METEOR Identifier:

409209

1.5.41. SKL Specification Version Number

Definition:

The version number of the SKL specification document used

Domain:

Value = *05.01*

Field Name:

`SpecVer`

1.5.42. State/Territory Identifier

Definition:

An identifier indicating the State or Territory responsible for the collection and submission of the SKL data file.

Domain:

1:

New South Wales

2:

Victoria

3:

Queensland

4:

South Australia

5:

Western Australia

6:

Tasmania

7:

Northern Territory

8:

Australian Capital Territory

(METEOR includes code 9, but that is not applicable to SKL)

Field Name:

`State`

METEOR Identifier:

790405

1.5.43. State/Territory Name

Domain:

Name used to identify the State/Territory

Field Name:

`StateName`

1.5.44. Target Population

Domain:

1:

Child and adolescent

2:

Older person

3:

Forensic

4:

General

5:

Youth

7:

Not applicable

Code 7 only applies to FTEORG usage, METEOR code 9 does not apply

Field Name:

`TargetPop`

METEOR Identifier:

682403

1.6. Virtual elements

1.6.1. AdmiAdmiCount

Base:

ADMI

Title:

ADMI Count at ADMI Level

SQL:

```
select State,
       RegId,
       OrgId,
       HospId,
       AdmiId,
       coalesce(Count, 0) as Count
  from ADMI
  left join (
        select State,
               RegId,
               OrgId,
               HospId,
               AdmiId,
               count(*) as Count
          from ADMI
         group by State,
                  RegId,
                  OrgId,
                  HospId,
                  AdmiId
       ) as tmpinner using (State, RegId, OrgId, HospId, AdmiId)
```

1.6.2. AdmiHasAdmi

Base:
ADMI
Title:
ADMI below ADMI
SQL:

```
select State,
       RegId,
       OrgId,
       HospId,
       AdmiId,
       Count
  from AdmiAdmiCount
 where Count > 0
```

1.6.3. AmbuAmbuCount

Base:
AMBU
Title:
AMBU Count at AMBU Level

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       AmbuId,
       coalesce(Count, 0) as Count
  from AMBU
  left join (
        select State,
               RegId,
               OrgId,
               ClusId,
               AmbuId,
               count(*) as Count
          from AMBU
         group by State,
                  RegId,
                  OrgId,
                  ClusId,
                  AmbuId
       ) as tmpinner using (State, RegId, OrgId, ClusId, AmbuId)
```

1.6.4. AmbuHasAmbu

Base:
AMBU
Title:
AMBU below AMBU
SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       AmbuId,
       Count
  from AmbuAmbuCount
 where Count > 0
```

1.6.5. ClusAmbuCount

Base:
CLUS
Title:
AMBU Count at CLUS Level

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       coalesce(Count, 0) as Count
  from CLUS
  left join (
        select State,
               RegId,
               OrgId,
               ClusId,
               count(*) as Count
          from AMBU
         group by State,
                  RegId,
                  OrgId,
                  ClusId
       ) as tmpinner using (State, RegId, OrgId, ClusId)
```

Rules:

- ClusBarren

1.6.6. ClusHasAmbu

Base:
CLUS
Title:
AMBU below CLUS
SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       Count
  from ClusAmbuCount
 where Count > 0
```

1.6.7. ClusHasResi

Base:
CLUS
Title:
RESI below CLUS

SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       Count
  from ClusResiCount
 where Count > 0
```

1.6.8. ClusResiCount

Base:
CLUS
Title:
RESI Count at CLUS Level
SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       coalesce(Count, 0) as Count
  from CLUS
  left join (
        select State,
               RegId,
               OrgId,
               ClusId,
               count(*) as Count
          from RESI
         group by State,
                  RegId,
                  OrgId,
                  ClusId
       ) as tmpinner using (State, RegId, OrgId, ClusId)
```

Rules:

- ClusBarren

1.6.9. HospAdmiCount

Base:
HOSP
Title:
ADMI Count at HOSP Level

SQL:

```
select State,
       RegId,
       OrgId,
       HospId,
       coalesce(Count, 0) as Count
  from HOSP
  left join (
        select State,
               RegId,
               OrgId,
               HospId,
               count(*) as Count
          from ADMI
         group by State,
                  RegId,
                  OrgId,
                  HospId
       ) as tmpinner using (State, RegId, OrgId, HospId)
```

1.6.10. HospHasAdmi

Base:
HOSP
Title:
ADMI below HOSP
SQL:

```
select State,
       RegId,
       OrgId,
       HospId,
       Count
  from HospAdmiCount
 where Count > 0
```

1.6.11. OrgAdmiCACount

Base:
ORG
Title:
ADMI Child and adolescent Count at ORG Level

SQL:

```
select State,
       RegId,
       OrgId,
       coalesce(Count, 0) as Count
  from ORG
  left join (
        select State,
               RegId,
               OrgId,
               count(*) as Count
          from ADMI
         where TargetPop = '1'
         group by State,
                  RegId,
                  OrgId
       ) as tmpinner using (State, RegId, OrgId)
```

1.6.12. OrgAdmiCount

Base:
ORG
Title:
ADMI Count at ORG Level
SQL:

```
select State,
       RegId,
       OrgId,
       coalesce(Count, 0) as Count
  from ORG
  left join (
        select State,
               RegId,
               OrgId,
               count(*) as Count
          from ADMI
         group by State,
                  RegId,
                  OrgId
       ) as tmpinner using (State, RegId, OrgId)
```

1.6.13. OrgAdmiForCount

Base:
ORG
Title:
ADMI Forensic Count at ORG Level

SQL:

```
select State,
       RegId,
       OrgId,
       coalesce(Count, 0) as Count
  from ORG
  left join (
        select State,
               RegId,
               OrgId,
               count(*) as Count
          from ADMI
         where TargetPop = '3'
         group by State,
                  RegId,
                  OrgId
       ) as tmpinner using (State, RegId, OrgId)
```

1.6.14. OrgAdmiGenCount

Base:
ORG
Title:
ADMI General Count at ORG Level
SQL:

```
select State,
       RegId,
       OrgId,
       coalesce(Count, 0) as Count
  from ORG
  left join (
        select State,
               RegId,
               OrgId,
               count(*) as Count
          from ADMI
         where TargetPop = '4'
         group by State,
                  RegId,
                  OrgId
       ) as tmpinner using (State, RegId, OrgId)
```

1.6.15. OrgAdmiOldCount

Base:
ORG
Title:
ADMI Older person Count at ORG Level

SQL:

```
select State,
       RegId,
       OrgId,
       coalesce(Count, 0) as Count
  from ORG
  left join (
        select State,
               RegId,
               OrgId,
               count(*) as Count
          from ADMI
         where TargetPop = '2'
         group by State,
                  RegId,
                  OrgId
       ) as tmpinner using (State, RegId, OrgId)
```

1.6.16. OrgAdmiYthCount

Base:
ORG
Title:
ADMI Youth Count at ORG Level
SQL:

```
select State,
       RegId,
       OrgId,
       coalesce(Count, 0) as Count
  from ORG
  left join (
        select State,
               RegId,
               OrgId,
               count(*) as Count
          from ADMI
         where TargetPop = '5'
         group by State,
                  RegId,
                  OrgId
       ) as tmpinner using (State, RegId, OrgId)
```

1.6.17. OrgAmbuCACount

Base:
ORG
Title:
AMBU Child and adolescent Count at ORG Level

SQL:

```
select State,
       RegId,
       OrgId,
       coalesce(Count, 0) as Count
  from ORG
  left join (
        select State,
               RegId,
               OrgId,
               count(*) as Count
          from AMBU
         where TargetPop = '1'
         group by State,
                  RegId,
                  OrgId
       ) as tmpinner using (State, RegId, OrgId)
```

1.6.18. OrgAmbuCount

Base:
ORG
Title:
AMBU Count at ORG Level
SQL:

```
select State,
       RegId,
       OrgId,
       coalesce(Count, 0) as Count
  from ORG
  left join (
        select State,
               RegId,
               OrgId,
               count(*) as Count
          from AMBU
         group by State,
                  RegId,
                  OrgId
       ) as tmpinner using (State, RegId, OrgId)
```

1.6.19. OrgAmbuForCount

Base:
ORG
Title:
AMBU Forensic Count at ORG Level

SQL:

```
select State,
       RegId,
       OrgId,
       coalesce(Count, 0) as Count
  from ORG
  left join (
        select State,
               RegId,
               OrgId,
               count(*) as Count
          from AMBU
         where TargetPop = '3'
         group by State,
                  RegId,
                  OrgId
       ) as tmpinner using (State, RegId, OrgId)
```

1.6.20. OrgAmbuGenCount

Base:
ORG
Title:
AMBU General Count at ORG Level
SQL:

```
select State,
       RegId,
       OrgId,
       coalesce(Count, 0) as Count
  from ORG
  left join (
        select State,
               RegId,
               OrgId,
               count(*) as Count
          from AMBU
         where TargetPop = '4'
         group by State,
                  RegId,
                  OrgId
       ) as tmpinner using (State, RegId, OrgId)
```

1.6.21. OrgAmbuOldCount

Base:
ORG
Title:
AMBU Older person Count at ORG Level

SQL:

```
select State,
       RegId,
       OrgId,
       coalesce(Count, 0) as Count
  from ORG
  left join (
        select State,
               RegId,
               OrgId,
               count(*) as Count
          from AMBU
         where TargetPop = '2'
         group by State,
                  RegId,
                  OrgId
       ) as tmpinner using (State, RegId, OrgId)
```

1.6.22. OrgAmbuYthCount

Base:
ORG
Title:
AMBU Youth Count at ORG Level
SQL:

```
select State,
       RegId,
       OrgId,
       coalesce(Count, 0) as Count
  from ORG
  left join (
        select State,
               RegId,
               OrgId,
               count(*) as Count
          from AMBU
         where TargetPop = '5'
         group by State,
                  RegId,
                  OrgId
       ) as tmpinner using (State, RegId, OrgId)
```

1.6.23. OrgClusCount

Base:
ORG
Title:
CLUS Count at ORG Level

SQL:

```
select State,
       RegId,
       OrgId,
       coalesce(Count, 0) as Count
  from ORG
  left join (
        select State,
               RegId,
               OrgId,
               count(*) as Count
          from CLUS
         group by State,
                  RegId,
                  OrgId
       ) as tmpinner using (State, RegId, OrgId)
```

Rules:

- OrgBarren

1.6.24. OrgHasAdmi

Base:
ORG
Title:
ADMI below ORG
SQL:

```
select State,
       RegId,
       OrgId,
       Count
  from OrgAdmiCount
 where Count > 0
```

1.6.25. OrgHasAdmiCA

Base:
ORG
Title:
ADMI Child and adolescent below ORG
SQL:

```
select State,
       RegId,
       OrgId,
       Count
  from OrgAdmiCACount
 where Count > 0
```

1.6.26. OrgHasAdmiFor

Base:
ORG
Title:
ADMI Forensic below ORG
SQL:

```
select State,
       RegId,
       OrgId,
       Count
  from OrgAdmiForCount
 where Count > 0
```

1.6.27. OrgHasAdmiGen

Base:
ORG
Title:
ADMI General below ORG
SQL:

```
select State,
       RegId,
       OrgId,
       Count
  from OrgAdmiGenCount
 where Count > 0
```

1.6.28. OrgHasAdmiOld

Base:
ORG
Title:
ADMI Older person below ORG
SQL:

```
select State,
       RegId,
       OrgId,
       Count
  from OrgAdmiOldCount
 where Count > 0
```

1.6.29. OrgHasAdmiYth

Base:
ORG
Title:
ADMI Youth below ORG

SQL:

```
select State,
       RegId,
       OrgId,
       Count
  from OrgAdmiYthCount
 where Count > 0
```

1.6.30. OrgHasAmbu

Base:
ORG
Title:
AMBU below ORG
SQL:

```
select State,
       RegId,
       OrgId,
       Count
  from OrgAmbuCount
 where Count > 0
```

1.6.31. OrgHasAmbuCA

Base:
ORG
Title:
AMBU Child and adolescent below ORG
SQL:

```
select State,
       RegId,
       OrgId,
       Count
  from OrgAmbuCACount
 where Count > 0
```

1.6.32. OrgHasAmbuFor

Base:
ORG
Title:
AMBU Forensic below ORG
SQL:

```
select State,
       RegId,
       OrgId,
       Count
  from OrgAmbuForCount
 where Count > 0
```

1.6.33. OrgHasAmbuGen

Base:
ORG
Title:
AMBU General below ORG
SQL:

```
select State,
       RegId,
       OrgId,
       Count
  from OrgAmbuGenCount
 where Count > 0
```

1.6.34. OrgHasAmbuOld

Base:
ORG
Title:
AMBU Older person below ORG
SQL:

```
select State,
       RegId,
       OrgId,
       Count
  from OrgAmbuOldCount
 where Count > 0
```

1.6.35. OrgHasAmbuYth

Base:
ORG
Title:
AMBU Youth below ORG
SQL:

```
select State,
       RegId,
       OrgId,
       Count
  from OrgAmbuYthCount
 where Count > 0
```

1.6.36. OrgHasClus

Base:
ORG
Title:
CLUS below ORG

SQL:

```
select State,
       RegId,
       OrgId,
       Count
  from OrgClusCount
 where Count > 0
```

1.6.37. OrgHasHosp

Base:
ORG
Title:
HOSP below ORG
SQL:

```
select State,
       RegId,
       OrgId,
       Count
  from OrgHospCount
 where Count > 0
```

1.6.38. OrgHasResi

Base:
ORG
Title:
RESI below ORG
SQL:

```
select State,
       RegId,
       OrgId,
       Count
  from OrgResiCount
 where Count > 0
```

1.6.39. OrgHasResiCA

Base:
ORG
Title:
RESI Child and adolescent below ORG
SQL:

```
select State,
       RegId,
       OrgId,
       Count
  from OrgResiCACount
 where Count > 0
```

1.6.40. OrgHasResiFor

Base:
ORG
Title:
RESI Forensic below ORG
SQL:

```
select State,
       RegId,
       OrgId,
       Count
  from OrgResiForCount
 where Count > 0
```

1.6.41. OrgHasResiGen

Base:
ORG
Title:
RESI General below ORG
SQL:

```
select State,
       RegId,
       OrgId,
       Count
  from OrgResiGenCount
 where Count > 0
```

1.6.42. OrgHasResiOld

Base:
ORG
Title:
RESI Older person below ORG
SQL:

```
select State,
       RegId,
       OrgId,
       Count
  from OrgResiOldCount
 where Count > 0
```

1.6.43. OrgHasResiYth

Base:
ORG
Title:
RESI Youth below ORG

SQL:

```
select State,
       RegId,
       OrgId,
       Count
  from OrgResiYthCount
 where Count > 0
```

1.6.44. OrgHospCount

Base:
ORG
Title:
HOSP Count at ORG Level
SQL:

```
select State,
       RegId,
       OrgId,
       coalesce(Count, 0) as Count
  from ORG
  left join (
        select State,
               RegId,
               OrgId,
               count(*) as Count
          from HOSP
         group by State,
                  RegId,
                  OrgId
       ) as tmpinner using (State, RegId, OrgId)
```

Rules:

- OrgBarren

1.6.45. OrgResiCACount

Base:
ORG
Title:
RESI Child and adolescent Count at ORG Level

SQL:

```
select State,
       RegId,
       OrgId,
       coalesce(Count, 0) as Count
  from ORG
  left join (
        select State,
               RegId,
               OrgId,
               count(*) as Count
          from RESI
         where TargetPop = '1'
         group by State,
                  RegId,
                  OrgId
       ) as tmpinner using (State, RegId, OrgId)
```

1.6.46. OrgResiCount

Base:
ORG
Title:
RESI Count at ORG Level
SQL:

```
select State,
       RegId,
       OrgId,
       coalesce(Count, 0) as Count
  from ORG
  left join (
        select State,
               RegId,
               OrgId,
               count(*) as Count
          from RESI
         group by State,
                  RegId,
                  OrgId
       ) as tmpinner using (State, RegId, OrgId)
```

1.6.47. OrgResiForCount

Base:
ORG
Title:
RESI Forensic Count at ORG Level

SQL:

```
select State,
       RegId,
       OrgId,
       coalesce(Count, 0) as Count
  from ORG
  left join (
        select State,
               RegId,
               OrgId,
               count(*) as Count
          from RESI
         where TargetPop = '3'
         group by State,
                  RegId,
                  OrgId
       ) as tmpinner using (State, RegId, OrgId)
```

1.6.48. OrgResiGenCount

Base:
ORG
Title:
RESI General Count at ORG Level
SQL:

```
select State,
       RegId,
       OrgId,
       coalesce(Count, 0) as Count
  from ORG
  left join (
        select State,
               RegId,
               OrgId,
               count(*) as Count
          from RESI
         where TargetPop = '4'
         group by State,
                  RegId,
                  OrgId
       ) as tmpinner using (State, RegId, OrgId)
```

1.6.49. OrgResiOldCount

Base:
ORG
Title:
RESI Older person Count at ORG Level

SQL:

```
select State,
       RegId,
       OrgId,
       coalesce(Count, 0) as Count
  from ORG
  left join (
        select State,
               RegId,
               OrgId,
               count(*) as Count
          from RESI
         where TargetPop = '2'
         group by State,
                  RegId,
                  OrgId
       ) as tmpinner using (State, RegId, OrgId)
```

1.6.50. OrgResiYthCount

Base:
ORG
Title:
RESI Youth Count at ORG Level
SQL:

```
select State,
       RegId,
       OrgId,
       coalesce(Count, 0) as Count
  from ORG
  left join (
        select State,
               RegId,
               OrgId,
               count(*) as Count
          from RESI
         where TargetPop = '5'
         group by State,
                  RegId,
                  OrgId
       ) as tmpinner using (State, RegId, OrgId)
```

1.6.51. RegAdmiCount

Base:
REG
Title:
ADMI Count at REG Level

SQL:

```
select State,
       RegId,
       coalesce(Count, 0) as Count
  from REG
  left join (
        select State,
               RegId,
               count(*) as Count
          from ADMI
         group by State,
                  RegId
       ) as tmpinner using (State, RegId)
```

1.6.52. RegAmbuCount

Base:
REG
Title:
AMBU Count at REG Level
SQL:

```
select State,
       RegId,
       coalesce(Count, 0) as Count
  from REG
  left join (
        select State,
               RegId,
               count(*) as Count
          from AMBU
         group by State,
                  RegId
       ) as tmpinner using (State, RegId)
```

1.6.53. RegHasAdmi

Base:
REG
Title:
ADMI below REG
SQL:

```
select State,
       RegId,
       Count
  from RegAdmiCount
 where Count > 0
```

1.6.54. RegHasAmbu

Base:
REG
Title:
AMBU below REG
SQL:

```
select State,
       RegId,
       Count
  from RegAmbuCount
 where Count > 0
```

1.6.55. RegHasResi

Base:
REG
Title:
RESI below REG
SQL:

```
select State,
       RegId,
       Count
  from RegResiCount
 where Count > 0
```

1.6.56. RegResiCount

Base:
REG
Title:
RESI Count at REG Level
SQL:

```
select State,
       RegId,
       coalesce(Count, 0) as Count
  from REG
  left join (
        select State,
               RegId,
               count(*) as Count
          from RESI
         group by State,
                  RegId
       ) as tmpinner using (State, RegId)
```

1.6.57. ResiHasResi

Base:
RESI
Title:
RESI below RESI
SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       ResiId,
       Count
  from ResiResiCount
 where Count > 0
```

1.6.58. ResiResiCount

Base:
RESI
Title:
RESI Count at RESI Level
SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       ResiId,
       coalesce(Count, 0) as Count
  from RESI
  left join (
        select State,
               RegId,
               OrgId,
               ClusId,
               ResiId,
               count(*) as Count
          from RESI
         group by State,
                  RegId,
                  OrgId,
                  ClusId,
                  ResiId
       ) as tmpinner using (State, RegId, OrgId, ClusId, ResiId)
```

1.7. Rules

1.7.1. AdmiAdmiNameMissing

Class:
Missing

Priority:
High
Message:
Missing data - AdmiName `$AdmiName.q`
Mark:
ADMI.AdmiName
Description:
Missing data - Admitted Patient Service Unit Name (AdmiName)
SQL:

```
select State,
       RegId,
       OrgId,
       HospId,
       AdmiId,
       AdmiName
  from ADMI
 where AdmiName is null
```

Data Elements:

- AdmiId
- AdmiName
- HospId
- OrgId
- RegId
- State

1.7.2. AdmiClosed

Class:
Skeleton
Priority:
High
Message:
Admi closed, historical `$hist_name` (Id: `$entityId`) no longer exists
Description:
Admitted Patient Service Unit Closed - A matching historical Admitted Patient Service Unit was not found in the current data

1.7.3. AdmiOpened

Class:
Skeleton
Priority:
High
Message:
Admi opened, `$name` (Id: `$entityId`) not in historical data

Description:
Admitted Patient Service Unit Opened - A matching Admitted Patient Service Unit was not found in the historical data

1.7.4. AdmiRenamed

Class:
Historical
Priority:
High
Message:
Admi renamed from `$hist_name.qt` to `$AdmiName.qt`
Mark:
ADMI.AdmiName
Description:
Admitted Patient Service Unit Renamed - Admitted Patient Service Unit Name differs between historical and current data
SQL:

```
select State,
       RegId,
       OrgId,
       HospId,
       AdmiId,
       ADMI.AdmiName,
       hist_entity.AdmiName as hist_name
  from ADMI
  join hist.ADMI as hist_entity using(State, RegId, OrgId, HospId, AdmiId)
 where not sloppy_match(ADMI.AdmiName, hist_entity.AdmiName)
```

Data Elements:

- AdmiId
- AdmiName
- HospId
- OrgId
- RegId
- State

1.7.5. AdmiTargetPopChanged

Class:
Historical
Priority:
High
Message:
Admitted Patient Service Unit TargetPop changed from `$hist_TargetPop` to `$TargetPop`
Mark:
ADMI.TargetPop

Description:

Target Population Changed - Target Population value for Admitted Patient Service Unit differs between historical and current data

SQL:

```
select State,
       RegId,
       OrgId,
       HospId,
       AdmiId,
       ADMI.TargetPop,
       hist_ADMI.TargetPop as hist_TargetPop
  from ADMI
  join hist.ADMI as hist_ADMI using(State, RegId, OrgId, HospId, AdmiId)
 where ADMI.TargetPop != hist_ADMI.TargetPop
```

Data Elements:

- AdmiId
- HospId
- OrgId
- RegId
- State
- TargetPop

1.7.6. AdmiTargetPopMissing

Class:

Missing

Priority:

High

Message:

Missing data - TargetPop `$TargetPop.q`

Mark:

ADMI.TargetPop

Description:

Missing data - Target Population (TargetPop)

SQL:

```
select State,
       RegId,
       OrgId,
       HospId,
       AdmiId,
       TargetPop
  from ADMI
 where TargetPop is null
```

Data Elements:

- AdmiId
- HospId
- OrgId

- RegId
- State
- TargetPop

1.7.7. AmbuAmbuNameMissing

Class:
Missing
Priority:
High
Message:
Missing data - AmbuName `$AmbuName.q`
Mark:
AMBU.AmbuName
Description:
Missing data - Ambulatory Service Unit Name (AmbuName)
SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       AmbuId,
       AmbuName
  from AMBU
 where AmbuName is null
```

Data Elements:
- AmbuId
- AmbuName
- ClusId
- OrgId
- RegId
- State

1.7.8. AmbuClosed

Class:
Skeleton
Priority:
High
Message:
Ambu closed, historical `$hist_name` (Id: `$entityId`) no longer exists
Description:
Ambulatory Service Unit Closed - A matching historical Ambulatory Service Unit was not found in the current data

1.7.9. AmbuOpened

Class:
Skeleton
Priority:
High
Message:
Ambu opened, `$name` (Id: `$entityId`) not in historical data
Description:
Ambulatory Service Unit Opened - A matching Ambulatory Service Unit was not found in the historical data

1.7.10. AmbuRenamed

Class:
Historical
Priority:
High
Message:
Ambu renamed from `$hist_name.qt` to `$AmbuName.qt`
Mark:
AMBU.AmbuName
Description:
Ambulatory Service Unit Renamed - Ambulatory Service Unit Name differs between historical and current data
SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       AmbuId,
       AMBU.AmbuName,
       hist_entity.AmbuName as hist_name
  from AMBU
  join hist.AMBU as hist_entity using(State, RegId, OrgId, ClusId, AmbuId)
 where not sloppy_match(AMBU.AmbuName, hist_entity.AmbuName)
```

Data Elements:

- AmbuId
- AmbuName
- ClusId
- OrgId
- RegId
- State

1.7.11. AmbuSectorChanged

Class:
Historical
Priority:
High

Message:
Ambulatory Service Unit Sector changed from `$hist_Sector` to `$Sector`
Mark:
AMBU.Sector
Description:
Sector Changed - Sector value for Ambulatory Service Unit differs between historical and current data
SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       AmbuId,
       AMBU.Sector,
       hist_AMBU.Sector as hist_Sector
  from AMBU
  join hist.AMBU as hist_AMBU using(State, RegId, OrgId, ClusId, AmbuId)
 where AMBU.Sector != hist_AMBU.Sector
```

Data Elements:

- AmbuId
- ClusId
- OrgId
- RegId
- Sector
- State

1.7.12. AmbuSectorMissing

Class:
Missing
Priority:
High
Message:
Missing data - Sector `$Sector.q`
Mark:
AMBU.Sector
Description:
Missing data - Sector (Sector)
SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       AmbuId,
       Sector
  from AMBU
 where Sector is null
```

Data Elements:

- AmbuId
- ClusId
- OrgId
- RegId
- Sector
- State

1.7.13. AmbuTargetPopChanged

Class:
Historical
Priority:
High
Message:
Ambulatory Service Unit TargetPop changed from `$hist_TargetPop` to `$TargetPop`
Mark:
AMBU.TargetPop
Description:
Target Population Changed - Target Population value for Ambulatory Service Unit differs between historical and current data
SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       AmbuId,
       AMBU.TargetPop,
       hist_AMBU.TargetPop as hist_TargetPop
  from AMBU
  join hist.AMBU as hist_AMBU using(State, RegId, OrgId, ClusId, AmbuId)
 where AMBU.TargetPop != hist_AMBU.TargetPop
```

Data Elements:

- AmbuId
- ClusId
- OrgId
- RegId
- State
- TargetPop

1.7.14. AmbuTargetPopMissing

Class:
Missing
Priority:
High

Message:
Missing data - TargetPop `$TargetPop.q`
Mark:
AMBU.TargetPop
Description:
Missing data - Target Population (TargetPop)
SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       AmbuId,
       TargetPop
  from AMBU
 where TargetPop is null
```

Data Elements:

- AmbuId
- ClusId
- OrgId
- RegId
- State
- TargetPop

1.7.15. ClusBarren

Class:
Barren
Priority:
High
Message:
CLUS has no AMBU or RESI records
Mark:
CLUS
Description:
Barren Cluster - has no associated Residential or Ambulatory service units
SQL:

```
select State,
       RegId,
       OrgId,
       ClusId
  from ClusAmbuCount
  join ClusResiCount using(State, RegId, OrgId, ClusId)
 where ClusAmbuCount.Count = 0
   and ClusResiCount.Count = 0
```

Data Elements:

- ClusId
- OrgId

- RegId
- State

Virtual Elements:

- ClusAmbuCount
- ClusResiCount

1.7.16. ClusClosed

Class:
Skeleton
Priority:
High
Message:
Clus closed, historical `$hist_name` (Id: `$entityId`) no longer exists
Description:
Service Unit Cluster Closed - A matching historical Service Unit Cluster was not found in the current data

1.7.17. ClusClusNameMissing

Class:
Missing
Priority:
High
Message:
Missing data - ClusName `$ClusName.q`
Mark:
CLUS.ClusName
Description:
Missing data - Service Unit Cluster Name (ClusName)
SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       ClusName
  from CLUS
 where ClusName is null
```

Data Elements:

- ClusId
- ClusName
- OrgId
- RegId
- State

1.7.18. ClusOpened

Class:
Skeleton
Priority:
High
Message:
Clus opened, `$name` (Id: `$entityId`) not in historical data
Description:
Service Unit Cluster Opened - A matching Service Unit Cluster was not found in the historical data

1.7.19. ClusRenamed

Class:
Historical
Priority:
High
Message:
Clus renamed from `$hist_name.qt` to `$ClusName.qt`
Mark:
CLUS.ClusName
Description:
Service Unit Cluster Renamed - Service Unit Cluster Name differs between historical and current data
SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       CLUS.ClusName,
       hist_entity.ClusName as hist_name
  from CLUS
  join hist.CLUS as hist_entity using(State, RegId, OrgId, ClusId)
 where not sloppy_match(CLUS.ClusName, hist_entity.ClusName)
```

Data Elements:

- ClusId
- ClusName
- OrgId
- RegId
- State

1.7.20. HospClosed

Class:
Skeleton
Priority:
High
Message:
Hosp closed, historical `$hist_name` (Id: `$entityId`) no longer exists

Description:
Hospital Closed - A matching historical Hospital was not found in the current data

1.7.21. HospHospNameMissing

Class:
Missing
Priority:
High
Message:
Missing data - HospName `$HospName.q`
Mark:
HOSP.HospName
Description:
Missing data - Hospital Name (HospName)
SQL:

```
select State,
       RegId,
       OrgId,
       HospId,
       HospName
  from HOSP
 where HospName is null
```

Data Elements:

- HospId
- HospName
- OrgId
- RegId
- State

1.7.22. HospOpened

Class:
Skeleton
Priority:
High
Message:
Hosp opened, `$name` (Id: `$entityId`) not in historical data
Description:
Hospital Opened - A matching Hospital was not found in the historical data

1.7.23. HospRenamed

Class:
Historical
Priority:
High

Message:
Hosp renamed from `$hist_name.qt` to `$HospName.qt`
Mark:
HOSP.HospName
Description:
Hospital Renamed - Hospital Name differs between historical and current data
SQL:

```
select State,
       RegId,
       OrgId,
       HospId,
       HOSP.HospName,
       hist_entity.HospName as hist_name
  from HOSP
  join hist.HOSP as hist_entity using(State, RegId, OrgId, HospId)
 where not sloppy_match(HOSP.HospName, hist_entity.HospName)
```

Data Elements:

- HospId
- HospName
- OrgId
- RegId
- State

1.7.24. HospSectorChanged

Class:
Historical
Priority:
High
Message:
Hospital Sector changed from `$hist_Sector` to `$Sector`
Mark:
HOSP.Sector
Description:
Sector Changed - Sector value for Hospital differs between historical and current data
SQL:

```
select State,
       RegId,
       OrgId,
       HospId,
       HOSP.Sector,
       hist_HOSP.Sector as hist_Sector
  from HOSP
  join hist.HOSP as hist_HOSP using(State, RegId, OrgId, HospId)
 where HOSP.Sector != hist_HOSP.Sector
```

Data Elements:

- HospId

- OrgId
- RegId
- Sector
- State

1.7.25. HospSectorMissing

Class:
Missing
Priority:
High
Message:
Missing data - Sector `$Sector.q`
Mark:
HOSP.Sector
Description:
Missing data - Sector (Sector)
SQL:

```
select State,
       RegId,
       OrgId,
       HospId,
       Sector
  from HOSP
 where Sector is null
```

Data Elements:
- HospId
- OrgId
- RegId
- Sector
- State

1.7.26. HrGenDtMissing

Class:
Missing
Priority:
High
Message:
Missing data - GenDt `$GenDt.q`
Mark:
HR.GenDt
Description:
Missing data - Data File Generation Date (GenDt)

SQL:

```
select State,
       GenDt
  from HR
 where GenDt is null
```

Data Elements:

- GenDt
- State

1.7.27. OrgBarren

Class:
Barren
Priority:
High
Message:
ORG has no CLUS or HOSP records
Mark:
ORG
Description:
Barren Organisation - has no associated Hospitals or Service Unit Clusters
SQL:

```
select State,
       RegId,
       OrgId
  from OrgHospCount
  join OrgClusCount using(State, RegId, OrgId)
 where OrgHospCount.Count = 0
   and OrgClusCount.Count = 0
```

Data Elements:

- OrgId
- RegId
- State

Virtual Elements:

- OrgClusCount
- OrgHospCount

1.7.28. OrgClosed

Class:
Skeleton
Priority:
High
Message:
Org closed, historical `$hist_name` (Id: `$entityId`) no longer exists

Description:
Organisation Closed - A matching historical Organisation was not found in the current data

1.7.29. OrgOpened

Class:
Skeleton
Priority:
High
Message:
Org opened, `$name` (Id: `$entityId`) not in historical data
Description:
Organisation Opened - A matching Organisation was not found in the historical data

1.7.30. OrgOrgNameMissing

Class:
Missing
Priority:
High
Message:
Missing data - OrgName `$OrgName.q`
Mark:
ORG.OrgName
Description:
Missing data - Organisation Name (OrgName)
SQL:

```
select State,
       RegId,
       OrgId,
       OrgName
  from ORG
 where OrgName is null
```

Data Elements:

- OrgId
- OrgName
- RegId
- State

1.7.31. OrgRenamed

Class:
Historical
Priority:
High
Message:
Org renamed from `$hist_name.qt` to `$OrgName.qt`

Mark:
ORG.OrgName
Description:
Organisation Renamed - Organisation Name differs between historical and current data
SQL:

```
select State,
       RegId,
       OrgId,
       ORG.OrgName,
       hist_entity.OrgName as hist_name
  from ORG
  join hist.ORG as hist_entity using(State, RegId, OrgId)
 where not sloppy_match(ORG.OrgName, hist_entity.OrgName)
```

Data Elements:

- OrgId
- OrgName
- RegId
- State

1.7.32. RegClosed

Class:
Skeleton
Priority:
High
Message:
Reg closed, historical `$hist_name` (Id: `$entityId`) no longer exists
Description:
Region Closed - A matching historical Region was not found in the current data

1.7.33. RegOpened

Class:
Skeleton
Priority:
High
Message:
Reg opened, `$name` (Id: `$entityId`) not in historical data
Description:
Region Opened - A matching Region was not found in the historical data

1.7.34. RegRegNameMissing

Class:
Missing
Priority:
High

Message:
Missing data - RegName `$RegName.q`
Mark:
REG.RegName
Description:
Missing data - Region Name (RegName)
SQL:

```
select State,
       RegId,
       RegName
  from REG
 where RegName is null
```

Data Elements:

- RegId
- RegName
- State

1.7.35. RegRenamed

Class:
Historical
Priority:
High
Message:
Reg renamed from `$hist_name.qt` to `$RegName.qt`
Mark:
REG.RegName
Description:
Region Renamed - Region Name differs between historical and current data
SQL:

```
select State,
       RegId,
       REG.RegName,
       hist_entity.RegName as hist_name
  from REG
  join hist.REG as hist_entity using(State, RegId)
 where not sloppy_match(REG.RegName, hist_entity.RegName)
```

Data Elements:

- RegId
- RegName
- State

1.7.36. RegngoNGONameMissing

Class:
Missing

Priority:
High
Message:
Missing data - NGOName `$NGOName.q`
Mark:
REGNGO.NGOName
Description:
Missing data - Non-Government Organisation Name (NGOName)
SQL:

```
select State,
       RegId,
       NGOId,
       NGOName
  from REGNGO
 where NGOName is null
```

Data Elements:

- NGOId
- NGOName
- RegId
- State

1.7.37. RegngoRenamed

Class:
Historical
Priority:
High
Message:
Regngo renamed from `$hist_name.qt` to `$RegngoName.qt`
Mark:
REGNGO.RegngoName
Description:
Region Mental Health Non-Government Organisation Renamed - Region Mental Health Non-Government Organisation Name differs between historical and current data
SQL:

```
select State,
       RegId,
       NGOId,
       REGNGO.RegngoName,
       hist_entity.RegngoName as hist_name
  from REGNGO
  join hist.REGNGO as hist_entity using(State, RegId, NGOId)
 where not sloppy_match(REGNGO.RegngoName, hist_entity.RegngoName)
```

Data Elements:

- NGOId
- RegId

- State

1.7.38. ResiClosed

Class:
Skeleton
Priority:
High
Message:
Resi closed, historical `$hist_name` (Id: `$entityId`) no longer exists
Description:
Residential Service Unit Closed - A matching historical Residential Service Unit was not found in the current data

1.7.39. ResiOpened

Class:
Skeleton
Priority:
High
Message:
Resi opened, `$name` (Id: `$entityId`) not in historical data
Description:
Residential Service Unit Opened - A matching Residential Service Unit was not found in the historical data

1.7.40. ResiRenamed

Class:
Historical
Priority:
High
Message:
Resi renamed from `$hist_name.qt` to `$ResiName.qt`
Mark:
RESI.ResiName
Description:
Residential Service Unit Renamed - Residential Service Unit Name differs between historical and current data
SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       ResiId,
       RESI.ResiName,
       hist_entity.ResiName as hist_name
  from RESI
  join hist.RESI as hist_entity using(State, RegId, OrgId, ClusId, ResiId)
 where not sloppy_match(RESI.ResiName, hist_entity.ResiName)
```

Data Elements:

- ClusId
- OrgId
- RegId
- ResiId
- ResiName
- State

1.7.41. ResiResiNameMissing

Class:
Missing
Priority:
High
Message:
Missing data - ResiName `$ResiName.q`
Mark:
RESI.ResiName
Description:
Missing data - Residential Service Unit Name (ResiName)
SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       ResiId,
       ResiName
  from RESI
 where ResiName is null
```

Data Elements:

- ClusId
- OrgId
- RegId
- ResiId
- ResiName
- State

1.7.42. ResiSectorChanged

Class:
Historical
Priority:
High
Message:
Residential Service Unit Sector changed from `$hist_Sector` to `$Sector`

Mark:
RESI.Sector
Description:
Sector Changed - Sector value for Residential Service Unit differs between historical and current data
SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       ResiId,
       RESI.Sector,
       hist_RESI.Sector as hist_Sector
  from RESI
  join hist.RESI as hist_RESI using(State, RegId, OrgId, ClusId, ResiId)
 where RESI.Sector != hist_RESI.Sector
```

Data Elements:

- ClusId
- OrgId
- RegId
- ResiId
- Sector
- State

1.7.43. ResiSectorMissing

Class:
Missing
Priority:
High
Message:
Missing data - Sector `$Sector.q`
Mark:
RESI.Sector
Description:
Missing data - Sector (Sector)
SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       ResiId,
       Sector
  from RESI
 where Sector is null
```

Data Elements:

- ClusId
- OrgId

- RegId
- ResiId
- Sector
- State

1.7.44. ResiTargetPopChanged

Class:
Historical
Priority:
High
Message:
Residential Service Unit TargetPop changed from `$hist_TargetPop` to `$TargetPop`
Mark:
RESI.TargetPop
Description:
Target Population Changed - Target Population value for Residential Service Unit differs between historical and current data
SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       ResiId,
       RESI.TargetPop,
       hist_RESI.TargetPop as hist_TargetPop
  from RESI
  join hist.RESI as hist_RESI using(State, RegId, OrgId, ClusId, ResiId)
 where RESI.TargetPop != hist_RESI.TargetPop
```

Data Elements:
- ClusId
- OrgId
- RegId
- ResiId
- State
- TargetPop

1.7.45. ResiTargetPopMissing

Class:
Missing
Priority:
High
Message:
Missing data - TargetPop `$TargetPop.q`
Mark:
RESI.TargetPop

Description:
Missing data - Target Population (TargetPop)
SQL:

```
select State,
       RegId,
       OrgId,
       ClusId,
       ResiId,
       TargetPop
  from RESI
 where TargetPop is null
```

Data Elements:

- ClusId
- OrgId
- RegId
- ResiId
- State
- TargetPop

1.7.46. StStateNameMissing

Class:
Missing
Priority:
High
Message:
Missing data - StateName `$StateName.q`
Mark:
ST.StateName
Description:
Missing data - State/Territory Name (StateName)
SQL:

```
select State,
       StateName
  from ST
 where StateName is null
```

Data Elements:

- State
- StateName

1.7.47. StngoNGONameMissing

Class:
Missing
Priority:
High

Message:

Missing data - NGOName `$NGOName.q`

Mark:

STNGO.NGOName

Description:

Missing data - Non-Government Organisation Name (NGOName)

SQL:

```
select State,
       NGOId,
       NGOName
  from STNGO
 where NGOName is null
```

Data Elements:

- NGOId
- NGOName
- State

1.7.48. StngoRenamed

Class:

Historical

Priority:

High

Message:

Stngo renamed from `$hist_name.qt` to `$StngoName.qt`

Mark:

STNGO.StngoName

Description:

State/Territory Mental Health Non-Government Organisation Renamed - State/Territory Mental Health Non-Government Organisation Name differs between historical and current data

SQL:

```
select State,
       NGOId,
       STNGO.StngoName,
       hist_entity.StngoName as hist_name
  from STNGO
  join hist.STNGO as hist_entity using(State, NGOId)
 where not sloppy_match(STNGO.StngoName, hist_entity.StngoName)
```

Data Elements:

- NGOId
- State

2. Submission and validation of SKL data files

Submission, delivery and validation of the 2026-27 SKL data (version 5.01) will occur through the Online Validator.

2.1. Timelines

Jurisdictions are requested to propose a file using the Online Validator by **3 January 2028**. The AIHW and Department of Health, Disability and Ageing are aiming to have fully reviewed and validated the file by **14 April 2028** in accordance with the schedule in Table 2.1, so structural changes are anticipated prior to the MHE file submissions.

Table 2.1 SKL 2026-27 issues validation

Progress point description	Responsibility	Completion Time	Completion date
Compliant file proposed for review	Jurisdictions		3 January 2028
Submitter comment on issues within the issue list	Jurisdictions	7 weeks	18 February 2028
Reviewer reply to issues within the issue list	AIHW	3 weeks	10 March 2028
Resolution of any remaining issues - validation process completed and submission finalised	AIHW/Jurisdiction discussion back and forth	5 weeks	14 April 2028

2.2. File type and naming convention

DAT files should be a single Fixed Format data file, with each record in the file being terminated with Carriage Return (CR) and Line Feed (LF) characters.

The data file will have the naming convention of *SKLSSSYYYYNNNNN.DAT* where:

- SKL denotes 'Mental Health Establishments Skeleton'
- SSS is the abbreviation for the State name, using the following convention:
  - New South Wales = NSW
  - Victoria = VIC
  - Queensland = QLD
  - Western Australia = WAU
  - South Australia = SAU
  - Tasmania = TAS
  - Australian Capital Territory = ACT
  - Northern Territory = NTE
- YYYY indicates the reporting year covered in the file, using the convention where financial years are abbreviated by referring to the last calendar year of the pair (for example, 2026-27 is identified as 2027)

- NNNNN represents an incremental batch number (leading zeros present).

Adherence to this approach requires that any resubmitted files should have a batch number greater than the file they replace. For example, the first MHE Skeleton data file submitted by the Australian Capital Territory covering the 2026-27 year would be named 'SKLACT202700001.DAT'.

2.3. Validation

Mental Health National Minimum Dataset (NMDS) validation is the process of reviewing and cleaning the mental health service data received from state and territory governments using the Online Validator a web based validation tool. The process includes:

- Checks on file upload, before the submitter chooses to propose the file for review
- The option for reviewers to accept or reject the proposed file for review
- Collaborative review and validation
    - Review and accept all issues in the issues list in consultation with jurisdictions as necessary.
    - Review data set reports (CMHC and RMHC) and raise additional issues with the jurisdiction as necessary.
    - Review historic trends reports (MHE) and author a supplementary report for jurisdiction review/response.
- A finalisation step, indicating that the submission process for that reporting cycle is complete.

2.3.1. Checks on file upload

The pre-submission checks ensure that the submitted data file structure is correct: that the data is in the correct layout, that there are no disallowed blank fields and no invalid characters. These checks ensure that each line of data is correctly formatted and aligns to the specifications.

2.3.2. Collaborative review and validation

Collaborative review and validation is the process of reviewing unusual trends in the files. Each collection is different and has its own structure, but the general process of validating is the same. In this process:

1. Submitters work though the issues list, either providing comments on known issues or updating and resubmitting the file. It is expected that most jurisdictions will need to submit updated files multiple times before validation is finalised. Note that jurisdictions must submit comments on issues before the AIHW is able to accept an issue.
2. Submitters and reviewers collaborate via the Online Validator, when more information is required on an issue. Each jurisdiction will have an AIHW staff member assigned as the lead validator, who will be in touch early in the validation period. Please ensure that any email communication is also Cc'd to the Mentalhealth@aihw.gov.au inbox in case of staff absences or changes in AIHW staffing.

3. The AIHW will review the 'data set reports' and will raise additional queries about any large fluctuations evident in these reports after investigating which regions, organisations and service units are causing them. The AIHW will endeavour to ensure that additional queries are limited to essential issues that impact the state-wide result.

2.4. Additional information and queries

AIHW, Department of Health, Disability and Ageing and Logicly staff are available to answer any queries in respect of changes to the Validator's MHE module and validation rules. In order to obtain a coordinated response, requests should be sent to the following parties simultaneously:

AIHW

Email: mentalhealth@aihw.gov.au

Logicly

Email: support@validator.com.au