4 Developing data elements

This section examines how to create a data element. A significant component of data development is the specification of data elements to standardise meaning and representation of the data to ensure consistency and comparability of data collected and information generated from a data collection.

During data development, a data element is specified for each concept that needs to be described or standardised within the data collection.

Before starting to develop data elements, it is important to understand the context of the collection and use for the data that are being described or standardised. This will provide the basis for the terminology to use. It also facilitates the selection or development of the metadata components (data element concept and value domain) that constitute the data element.

4.1 The components of a data element

A modified ISO/IEC 11179 model used to describe a data element and its supporting components is provided below.

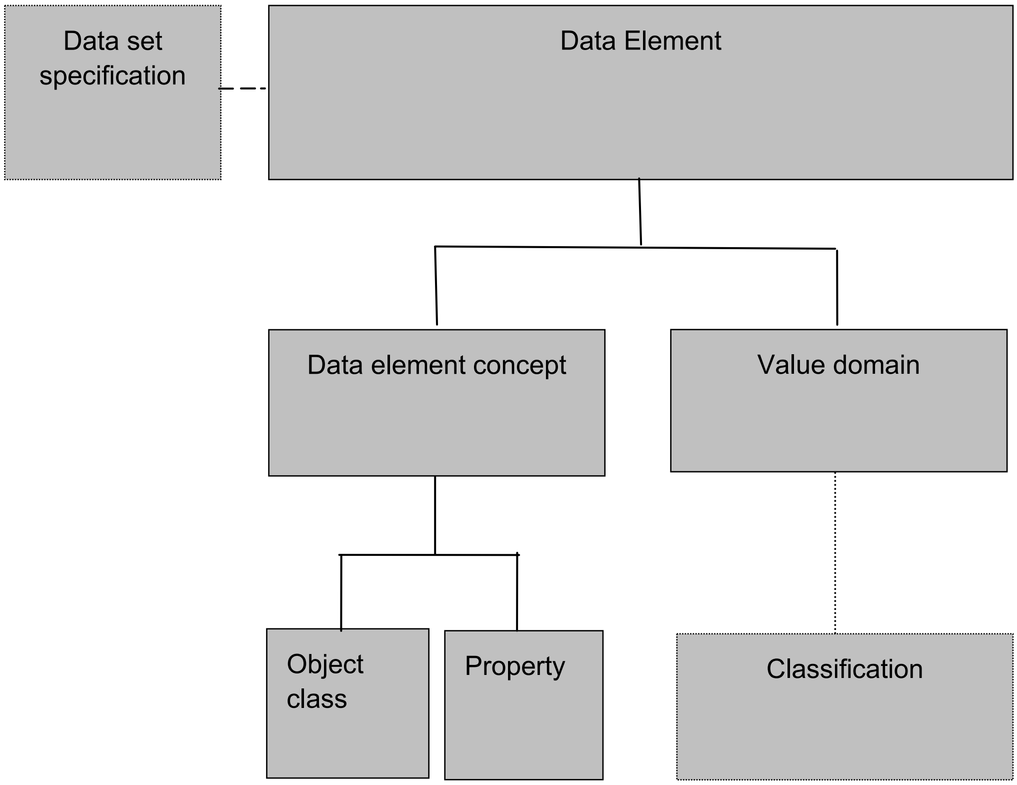

Figure 4: Model of data element and supporting components

A data element is made up of component parts. A data element is composed of a data element concept in association with a value domain. While the meaning of the data element is essentially captured within the data element concept, the value domain specifies its representation.

A value domain may itself be part of a classification, or a classification may provide the basis for value meanings of a value domain. Many examples of classifications exist, such as ICD-10-AM and the Australian Standard Classification of Religious Groups. Classifications such as these are official terminological systems, recognised and endorsed by national or international bodies that are used to classify data.

A data element concept is composed of an object class and a property.

A collection of data elements used to describe data for a specific purpose is a Data Set Specification (DSS). A NMDS is a special type of DSS—one consisting of a minimum set of data elements that are mandated for national collection. For example, the Elective Surgery Waiting Times (census data) NMDS contains the following data elements:

- Census date
- Clinical urgency
- Establishment identifier
- Extended wait patient
- Indicator procedure
- Listing date for care
- Overdue patient
- Surgical specialty
- Waiting time at a census date.

It is possible for a data element to be a component of more than one data set. For example, the data element 'Establishment identifier' is also a component of the Public Hospital Establishments NMDS.

A DSS specifies the conditions under which the data elements are collected. A DSS can define the sequence in which data elements are included, whether they are mandatory, what verification rules should be employed and the characteristics of the collection (for example, its scope).

4.1.1 Object Class

An object class is the identified 'thing' of interest for which the data developer is seeking to collect and store data. An object class has explicit boundaries, properties or attributes and meaning. Examples of an object class include car, person, household, employee, purchase order, client, hospital, agency, housing assistance agency, event and episode of admitted patient care.

Depending on what information is sought and what questions need to be answered, the broad entities (such as person, hospital, agency or event) may be further specialised or sub-typed to refine the 'thing' being described. For example, if a data developer were interested in grouping the population by sex, then 'person' can be further sub-typed into 'female' and 'male'. The sub-type becomes the Object Class.

A single object class can be sub-typed to more clearly describe the 'thing' of interest, where a greater degree of specialisation is required. A single object class can be sub-typed on more than one 'dimension'.

EXAMPLE—'Person (object class)'

sub-typed by age group

Child

Adult

sub-typed by Sex

Female

Male

EXAMPLE—'Hospital (object class)'

(sub-typed by primary funding source)

Public hospital

Private hospital

EXAMPLE—'Hospital'

(sub-typed by teaching status)

Teaching hospital

Non-teaching hospital

It is possible that during the course of a data development exercise other object classes are identified in addition to the ones already identified.

Question to ask when trying to identify the object class:

What is the 'thing' being described for which information is required?

4.1.2 Property

Having identified the object class, the next step is to identify the attributes, characteristics or aspects of the object class that the data developer is interested in describing or understanding. A property is the attribute, characteristic or aspect common to all members of the object class that is of interest.

For example, when considering the object class 'Client' the data developer may be interested in 'date of birth' or the 'sex' or 'postcode', and so on. For the object class 'Public hospital' the developer may be interested in the 'Identifier', 'Capital expenditure', 'Full-time equivalent staff', and so on.

In some instance it may not be easy to differentiate between a property and an object class. For example, an 'Identifier' may be defined as a property of a 'Person' (object class = person, property = identifier). In another context, the object class could be 'Identifier' and the property could be 'Number or designation', 'Name', 'Issuer', and so on.

The identification of object classes and properties is therefore contextual.

Questions to ask when trying to identify and define the property/ies.

What aspect, characteristic, property or attribute of the 'thing' does one want to describe? What does one need to know about the object class?

4.1.3 Data Element Concept (DEC)

A DEC is a concept that can be represented in the form of a data element, described independently of any particular representation. A data element concept is in fact a data element without representation.

A data element concept is a concept created by the union of an object class and a property. That is, one object class and one property are joined to form a data element concept.

Object class + Property = Data Element Concept

For example, if we are interested in the concept of whether a person lives alone or with others, the object class 'Person' could be associated with the property named 'Living Arrangement' in order to create a data element concept 'Person—living arrangement'.

As a data element, this DEC could be represented as:

Code	Description
1	lives alone
2	lives with others

Other examples of data element concepts include the day a person was born ('Person–date of birth') and the colour of a vehicle ('Vehicle–colour'). The basic characteristic of the DEC is the ability to define what we want to know, without the need to specify exactly how it should be represented.

If an existing data element is to be re-used, it is crucial that the concept that the data element needs to define is the same as that of the existing data element; otherwise a new data element should be created. In cases where the representation of the data are different for different data sets, it would be necessary to create two separate data elements. For example, an adult data set would specify measurement of a person's height in metres, while a neonatal data set would require a data element that specifies measurement in centimetres. In this case, conceptually the two data elements are the same, but, because the concept is represented differently, separate data elements are required.

Understanding the concept is crucial and this is why the data element concept and its component object class and property must be clearly defined in accordance with agreed standards in order to facilitate comparison.

4.1.4 Value Domain

A value domain provides the valid values and representation for the concept defined by a data element. The representational component of a data element is about the permitted values a data element may use. The set of these permitted values for a data element is called a value domain (VD). Each data element is only associated with one value domain.

A value domain can either be enumerated, where the value domain is specified by a list of all its permissible values (for example, 1=lives alone, 2 = lives with others, and so on), or non-enumerated, where the value domain is specified by a description rather than a list of all permissible values (for example, where values are specified by a range of values, such as age, height, weight, and so on).

An enumerated value domain contains a list of all its values and their associated value meanings. Each value and meaning pair is called a permissible value.

One of the first things to consider when developing data elements is how the concept represented by the data element is to be represented. The representation describes the form of the data, including a value domain, data type, representation class (optionally), format and, if necessary, a unit of measurement.

For example, the data element representing the concept 'date of commencement of service' may have dates in the format DDMMYYYY as valid values. This is its value domain. Or the data element representing the concept 'annual household income' may have the set of non-negative integers with units of Australian dollars, as a set of valid values. This is its value domain.

When selecting or developing value domains, it is important to ensure that they are consistent and mappable to national or international data standards, where these exist.

Where permissible values implement a classification or code set, the edition or version of the classification or code set should be referenced. Specifying the revision, edition or year of the classification enables consistent and comparable data collection, which would otherwise be left to assumptions being made about which version of the classification was or is used. For example, the edition or version of the classification should be referenced as part of the name of the value domain, such as 'Country code (SACC 1998) NNNN'

Questions to ask to help determine the value domain.

How do we want to represent the concept? What are the possible values that the concept can have?

Granularity

Depending on its purposes, value domains may be expressed to varying degrees of granularity. For example, the value domain for 'Eastern Asian Language Spoken' can be expressed as:

Eastern Asian Language Spoken

Value	*Value Meaning*
71	*Chinese*
72	*Japanese*
73	*Korean*
79	*Other Eastern Asian languages*

or

Eastern Asian Language Spoken

Value	*Value Meaning*
Chinese	
7101	*Cantonese*
7102	*Hakka*
7103	*Hokkien*
7104	*Mandarin*
7105	*Teochew*
7106	*Wu*
7199	*Other Chinese languages (includes Chang Chow, Hunan, Kan)*
Japanese	
7201	*Japanese*
Korean	
7301	*Korean*
Other Eastern Asian languages	
7999	*Other Eastern Asian languages (includes Ainu, Bhotia, Tibetan)*

The second Eastern Asian Language spoken value domain is expressed to a finer level of granularity. It is important that it is possible to map between value domains to ensure comparability, where this is required.

Alignment of requirements identified through the combined top-down and bottom-up approach to data development is frequently achieved through mapping of varying degrees of granularity of terms to recognised classifications. More detailed information is usually preferred for primary purposes, whereas less granular, but more stable, terms are often required for statistical purposes.

Compliance and consistency

In order to achieve semantic interoperability when data are exchanged between two parties or systems, the data must meet the data standards that are agreed to by the parties involved. Data exchanged between two parties is considered to be:

- **compliant**—when it meets all the requirements of the agreed data standard
- **consistent**—if the definition of the data element is the same as the agreed standard, but the Data type, Representation class, Format, Maximum field size or Data Domains are

different, and the data are still convertible to the agreed data standard without loss of the meaning.

Examples are provided below using the permissible values for the Marital Status data element. It should be noted that compliance and consistency are not just achieved when comparing domain values. In this example, the permissible values for the data standard for Marital Status are:

1	Never married
2	Widowed
3	Divorced
4	Separated
5	Married (including de facto)
9	Not stated/inadequately described

Data that uses only these coding categories, numbers and labels would be considered compliant with the data standard.

Examples of consistent and inconsistent values are:

Consistent with data standard ☑		Consistent with data standard ☑		Inconsistent with data standard ☒	
S	Never married	1	Never married	s	Single
W	Widowed	2	Widowed	w	Widowed
D	Divorced	3	Divorced		
A	Separated	4	Separated	a	Separated or Divorced
M	Married (including de facto)	5	Married (excluding de facto)	m	Married
		6	De facto		
Z	Not stated/inadequately described	9	Not stated/inadequately described		
Although the codes are not compliant with the data standard, they can be mapped (that is, converted) directly to the standard codes. A data element using these permissible values could be considered ‘consistent’ with the data standard.		Although codes 5 and 6 are not compliant with the data standard, the data are still ‘consistent’. Code 5 and code 6 data can be mapped to code 5 of the standard data domain.		‘Single’ is not the same as ‘Never married’ and may be misconstrued for persons who are divorced and now ‘single’ or for persons in a de facto relationship. Code ‘a’ cannot be mapped to the original data domain as it combines two different standard codes in one. ‘Married’ does not include ‘de facto’, which may lead to ’de facto’ being recorded under any of the other codes.	

4.1.5 Classification

Enumerated value domains may be a subset of a classification, such as ICD-10-AM. A classification ‘is a set of discrete exhaustive and mutually exclusive observations which can be assigned to one or more variables to be measured in the collation and/or presentation of data' (UN Glossary 2006).

Many examples of classifications exist, for example, ICD-10-AM, *the Australian Standard Classification of Religious Groups* and ICPC2+. Classifications such as these are used to classify data and are official terminological systems, recognised and endorsed by a national or international body.

Some classifications are updated more frequently than others. For example, the Australian Standard Geographical Classification (ASGC) is normally updated annually. The International Statistical Classification of Diseases and Related Health Problems, Tenth Revision, Australian Modification (ICD-10-AM), which is currently in its 5th edition, is updated every two years. Where permissible values implement a classification or code set, the edition or version of the classification or code set should be referenced. Specifying the revision, edition or year of the classification enables consistent and comparable data collection, which would otherwise be left to assumptions being made about which version of the classification was or is used.

Classifications endorsed for use within the Australia health sector belong to the Australian Family of Health and Related Classifications. The development of the Australian Family of Health and Health Related Classifications is managed by the Statistical Informational Management Committee (SIMC) and the Health Data Standards Committee (HDSC) on behalf of the National Health Information Management Principal Committee (NHIMPC).

4.2 Formulating data elements

A data element is the basic unit of identifiable and definable information. It is composed of a data element concept in association with a value domain. While the meaning of the data element is essentially captured within the data element concept, the value domain specifies its representation.

Data Element Concept (DEC) + Value Domain = Data Element

For example, if one is seeking to collect information about 'the height of people', then one could create a data element to capture this basic unit of information. The data element definition might be 'A person's height in metres, measured standing in bare feet'.

A data element concept can be represented by more than one data element. In the example above, the Data Element Concept would be 'A person's height'. This could then be instantiated as height in metres, millimetres, feet or inches. Each of these possibilities could be instantiated as a separate data element that measures a person's height.

Note: A common misconception is that a data element is a code set. This is not correct. A data element is a concept that is represented with a set of values. It is this set of values (codes) that are used for completing data entry screens or manual forms. While the codes are what most people see, they need to be supported by underlying concept definitions and information about the code's use.

4.2.1 Formulating data elements – an example

Let us reconsider the 'Living arrangement' example shown below.

EXAMPLE—'Living Arrangement'

Definition:	Whether a person usually resides alone or with others	
Data type:	Numeric	
Maximum Size:	1	
Data Domain:	1	Lives alone
	2	Lives with others
	9	Not stated / Inadequately described
Guide for use:	This item does not seek to describe the quality of the arrangements, but merely the fact of the arrangement. It is recognised that this item may change on a number of occasions during the course of an episode of care.	

Identifying the data element concept

The data element concept is the conceptual information about the data element. In this example, the 'thing' that the data relates to is the person or client (that is, the object class), depending on the terminology used in the specific context. The characteristic of interest of the person (or client) is the living arrangement of the person (that is, the property). The data element concept may be identified by the concatenation of the object class and property, which in this case is Person–living arrangement. In other words, the conceptual information of interest is the person's living arrangement, which in this case is defined as whether a person usually resides alone or with others.

Identifying the value domain

In the example above, the only characteristic of interest is whether a person lives alone or with others and the data developer wants to represent this by single digit numeric (N) codes.

Defining the data element

The resulting data element would consist of the data element concept 'Person–living arrangement' and it would be represented using a numeric code. The data element would be defined as 'whether a person usually resides alone or with others, as represented by a code'.

The name of the data element is dependent on the registry's business rules. For example, in METeOR, the name of the data element would be Person–living arrangement, code N.

Data elements should describe one concept and be appropriately represented

Where data elements are use to standardise data, a data element that describes a particular concept should not be represented or be described by values that do not represent the concept. For example, the concept 'type of fish living in an aquarium' is a totally different concept from 'the presence of fish living in an aquarium'. The former might represent data such as 'guppies', 'mollies', 'goldfish', 'swordfish', 'angel fish'…'. The latter might represent data such as 'There are no fish in the aquarium'. Neither data element on its own should be used to represent both sets of data.

Right representation		**Wrong Representation**	
The type of fish in an aquarium		The type of fish in an aquarium	
1	guppies	0	no fish
2	mollies	1	guppies
3	goldfish	2	mollies
4	swordfish	3	goldfish
5	angel fish	4	swordfish
8	other	5	angel fish
		8	other

Where information is required about both the presence of fish in the aquarium and the type of fish present, this should be represented by two separate data elements

Right representation		**Right Representation**	
The type of fish in an aquarium		The presence of fish in an aquarium	
1	guppies	1	present
2	mollies	2	absent
3	goldfish		
4	swordfish		
5	angel fish		
8	other		

Note: For the purpose of data collection and to minimise the burden on collectors the mixing of concepts is acceptable on collection forms as follows:

Types of fish in an aquarium

0	no fish
1	guppies
2	mollies
3	goldfish

4.3 Guidelines for good data standards

This section provides guidelines for creating good data standards, illustrated where possible using good and poor examples.

4.3.1 Name—guidelines

All data standards items in a metadata registry should have at least one name. Each registry establishes its own naming convention. ISO/IEC 11179-5 gives principles for naming data elements. It does not specify a mandatory naming convention.

The name should be

- unique in the registry for the metadata item type
- stated in the singular
- a reflection the concept being defined
- avoid using abbreviations or acronyms other than those widely accepted

4.3.2 Definition—guidelines

All data standards should have a definition. There are usually no 'wrong' definitions, just good ones and poor ones. ISO 11179-4 (2004) *Formulation of data definitions* has been reproduced in part to show the difference between a good definition and a poor definition. These include mandatory and recommended rules.

Mandatory rules

A definition must:

1. be unique within a registry and within the context of the metadata item

A data definition must be unique within a specific context within any registry and for the registration authority in which it appears. One or more characteristics expressed in the definition must differentiate its concept from other concepts.

2. be stated in the singular

The concept expressed by the data definition should be expressed in the singular. (An exception to this guideline is made where the concept itself is plural.)

EXAMPLE — 'Country name'

Good definition: *The commonly known short name of a country.*

Poor definition: *The commonly known short name of countries.*

REASON *– The poor definition uses the plural word 'countries', which is ambiguous, as it implies that a 'Country name' could refer to more than one country.*

3. state what the concept is, not what it is not

When constructing definitions, the concept cannot be defined exclusively by stating what the concept is not.

EXAMPLE —'Country short name'

Good definition*: The ISO recognised short name of a country.*

Poor definition: *The ISO recognised name that is not the long name of a country.*

REASON *– The poor definition does not specify what is included in the meaning of the data.*

4. be stated as a descriptive phrase or sentence

A phrase is necessary (in most languages) to form a precise definition that includes the essential characteristics of the concept. Simply stating one or more synonym(s) is insufficient. Simply restating the words of the name in a different order is insufficient. If more than a descriptive phrase is needed, use complete, grammatically correct sentences.

EXAMPLE —'Country name'

Good definition: *The commonly known short name that identifies a country.*

Poor definition: *Name of a country.*

REASON – *The poor definition does not describe the concept, that this is the short name, not an expanded or long name.*

5. contain only commonly understood abbreviations

Understanding the meaning of an abbreviation, including acronyms and initials, is usually confined to a certain environment. In other environments the same abbreviation can cause misinterpretation or confusion. Therefore, to avoid ambiguity, full words, not abbreviations, should be included in the definition.

Exceptions to this requirement may be made if an abbreviation is commonly understood such as 'i.e. ' and 'e.g.' or if an abbreviation is more readily understood than the full form of a complex term, and has been adopted as a term in its own right such as 'radar' standing for 'radio detecting and ranging'. All acronyms must be expanded on the first occurrence.

EXAMPLE —'Unit of Density Measurement'

Good definition: *The code that represents the unit for measuring mass per unit (m.p.u.) volume.*

Poor definition: *The code that represents the unit for measuring the m.p.u. volume.*

REASON— *m.p.u. is not a common abbreviation, and its meaning may not be understood by some users.*

6. be expressed without embedded definitions

The definition of a second data element or related concept should not appear in the definition of the primary data element.

EXAMPLE — 'Severe Hypoglycaemia History'

Good definition: *A code that represents whether a person has had severe hypoglycaemia in the last 12 months.*

Poor definition: *A code that represents whether a person has had severe hypoglycaemia, which is defined as hypoglycaemia requiring assistance from another party, in the last 12 months.*

REASON *– The poor definition contains a concept definition for severe hypoglycaemia, which should be included in a glossary.*

Recommended rules

A data definition should:

1. state the essential meaning of the concept

All primary characteristics of the concept represented should appear in the definition. The inclusion of non-essential characteristics should be avoided.

EXAMPLE—'Date of birth'

Good definition: *The date a person was born.*

Poor definition: *The date of birth of the person as stated in the birth certificate.*

REASON *– The poor definition includes extraneous material. How or where the information is captured can be stated in the collection methods. This information does not serve to define the concept.*

2. be precise and unambiguous

The exact meaning and interpretation of the defined concept should be apparent from the definition. A definition should be clear enough to allow for only one possible interpretation.

EXAMPLE —'Country code'

Good definition: *The 2-character alphabetic code assigned by the International Standard Organisation (ISO) 3166-1 to represent a country.*

Poor definition: *The code that represents a country.*

REASON *– Codes are assigned by ISO 3166-1:1997, the ABS, Standards Australia, and so on. Some codes are alphabetic and others are numeric and the number of characters can also vary. The poor definition is imprecise, making it difficult to determine the source of the code.*

3. be concise

The definition should be brief and comprehensive. Extraneous qualifying phrases, such as, 'for the purpose of this metadata registry' or 'terms to be described' should be avoided.

EXAMPLE—'Family Name'

Good definition: *A name shared in common to all members of a family, as distinguished from each member's given name.*

Poor definition: A name shared in common to all members of a family for purposes of identification, as distinguished from each member's given name.

REASON – The poor definition contains the extraneous qualifying phrase 'for purposes of identification'.

4. be able to stand alone

The meaning of the concept should be apparent from the definition. Additional explanations or references should not be necessary for understanding the meaning of the definition.

EXAMPLE—'School Location City Name'

Good definition: *Name of the city where a school is situated.*

Poor definition: *See "school site".*

REASON— *The poor definition does not stand alone, it requires the aid of a second definition (school site) to understand the meaning of the first definition.*

5. avoid circular reasoning

Two definitions should not be defined in terms of each other. A definition should not use another concept's definition as its definition.

EXAMPLE— two data elements with poor definitions:

a) Employee ID Number – Number assigned to an employee.

b) Employee – Person corresponding to the employee ID number.

REASON— Each definition refers to the other definition for its meaning. Accordingly, the meaning is not given in either definition.

6. use the same terminology and consistent logical structure for related definitions

A common terminology and syntax should be used for similar or associated definitions. Where the terminology and syntax are not the same, a user might assume that there is an implied difference between related definitions.

EXAMPLE—Good Consistency

The code that represents the method to determine the vertical coordinate.

The name of the method used to determine the vertical coordinate.

The name of the method used to determine the horizontal coordinates.

EXAMPLE—Poor Consistency

The name of the method used to determine the horizontal coordinates

The code that represents the method used to determine the latitude and longitude

REASON *– As the terminology is different (horizontal coordinates versus latitude and longitude), the registry user might assume that the different terms have a somewhat different meaning, even though they are simply different representations of the same concept.*

Data element concept definitions versus data element definitions

The definition of a data element concept should not make any reference to a specific representation. However, a data element definition must make reference to the specific representation.

EXAMPLE—Data Element Concept Definition

Data element concept: *'Job Grade Maximum Salary Amount'*

Definition: *The maximum salary permitted for the associated job grade.*

Note: The data element concept makes no reference to a specific value domain.

EXAMPLE—Data Element Definition

Data element 1*: 'European Job Grade Maximum Salary Amount'*

Definition: *The maximum salary permitted for the associated job grade expressed in Euros.*

Data element 2: *'U.S. Job Grade Maximum Salary Amount'*

Definition: *The maximum salary permitted for the associated job grade expressed in U.S. dollars.*

Note: Data element definitions may refer to explicit values domains, since this may be all that distinguishes two data elements.

Since the data element definition always includes representation, as a guide, the phrase that defines the data element should begin (or end) by stating the representation class for the data element and its value domain. The definite article 'the' is used, because the definition refers to a one instance of a data value.

Name:	*The name of …*
Code:	*The code that represents…*
or	*…as represented by a code*
Number:	*The number assigned by …*
or	*The number that represents*
or	*…as represented by a number*
Measure:	*The measure of the (area, mass, distance)…*
Quantity:	*The (sum, amount, capacity) of…*
Date:	*The date expressed as DDMMYYYY when …*

4.3.3 Context – guidelines

Data standards may have a context in which the definition has meaning. If the context is left blank, this will imply that the meaning imparted by the definition is valid in all contexts.

The context could be the setting in which data collection or use is valid (for example, juvenile justice or intensive care), or it could be a whole sector of service or care (such as custodial services or admitted patient care), or it could be more general, covering the whole of the health sector, or across service sectors.

Some examples of appropriate use of the context include:

- Public health
- a community aged care program
- a supported accommodation assistance program
- an emergency department.

Information about why the data element is important for collection should not be included in the context—for example, 'this item is collected for the analysis of outcome by treatment'. The purpose of justification for a data element may be included in the 'Comments' attribute or may be more appropriate as part of the data set specification (DSS).

4.3.4 Value domain—guidelines

The definition of the value domain can be plural, because it encompasses all permissible values for that domain. This is in contrast to the definition of the object class, property, data element concept and data element, where the concept is expressed in the singular

FOR EXAMPLE

The set of permissible values for……….

or

The code set representing………

Associated with a value domain are other attributes that make it easy to distinguish among data elements in a metadata registry. These include representational class, format, data type, maximum length, and so on, and are useful to help differentiate between data elements. For example, a data element categorised with a data type of 'string' is different from an element categorised as 'number'. It will not make sense to compare their contents or perform calculations. It is therefore useful to convey these representational attributes of a data element.

Representation class

Examples of representation class values and their associated meaning

Value	Meaning
Average	A numeric value representing an arithmetic mean
Code	A system of valid symbols that substitute for longer values
Date	A numeric value representing a calendar date (that is, day, month and year) or recognised part of a calendar date (that is, day, month, and/or year)
Identifier	A value which establishes identity
Percentage	Parts per hundred
Ratio	An expression of the quantity of one substance or entity in relation to that of another
Text	An unformatted, descriptive value
Time	A numeric value representing a specific instance in time
Total	A numeric value representing the sum of a set of values or an entire quantity (including monetary)

Data types

Examples of data types values and their associated meaning

Value	Meaning
Boolean	A binary value expressed using a string e.g. true or false.
Currency	A numeric value expressed using a particular medium of exchange.
Date/Time	A specific instance of time expressed in numeric form.
Number	A sequence of numeric characters which may contain decimals, excluding codes with ‘leading’ characters e.g. ‘01’,’02’,’03’.
String	A sequence of alphabetic and/or numeric characters, including ‘leading’ characters e.g. ‘01’,‘02’,‘03’.

Format

Examples of format values and their associated meaning

Value	Valid character range
A	Alphabetic character set: contains the letters a-z and A-Z and may contain special characters, but not numeric characters
N	Numeric character set: contains whole and decimal numbers and may contain special characters, but not alphabetic characters
X	Alphanumeric character set: contains alphabetic and numeric characters, and may contain blank characters
D	A numeric character representing a number of days
M	A numeric character representing a number of months
Y	A numeric character representing a number of years
H	Any numeric character representing a number of hours
M	Any numeric character representing a number of minutes
S	Any numeric character representing number of seconds

Unit of measure

Unit of measure—some value domains require that values for a data element be measured in only one unit (for example, height in centimetres). This attribute contains the name of the unit of measure for all data values for the value domain.

Examples of Units of measure classified by measure

Measurement	Unit of measure name	Unit of measure symbol
Concentration	Microgram per litre	μg/l
	Milligram per 24-hour period	mg/24h
	Nanogram per decilitre	ng/dl
Currency	Australian currency	AU$
Length	Centimetre	cm
	Millimetre	mm
Temperature	Degree Celsius	°C
Time	Second	s
	Minute	min
	Hour	h
	Day	d
Weight	Gram	g

4.3.5 Permissible values—guidelines

The value domain is specified by a list of all its permissible values. In a value domain, the permissible values must be exhaustive within the value domain and mutually exclusive with the value domain.

Exhaustive within the value domain

In the example below, the permissible values listed is not a full set of 'living thing'. Plants, microscopic animals (including some things that are both plant and animal), viruses and bacteria are missing.

Example—'Living thing'

Permissible values	Value	Meaning
	1	Fish
	2	Bird
	3	Mammal
	4	Reptile

'Other' should be used to ensure an exhaustive set of permissible values. See the example below. The use of the term 'Other' by itself actually signifies 'other living thing' or 'other living thing not elsewhere classified (more often recorded as "Living thing nec")'.

Permissible values	Value	Meaning
	1	Fish
	2	Bird
	3	Mammal
	4	Reptile
	8	**Other**

Mutually exclusive within the value domain

In the example below, the permissible values listed are not mutually exclusive. Marsupials are a subset of mammals. This means that a kangaroo could be counted twice, as both a mammal and a marsupial. Codes 3 and 5 are therefore not mutually exclusive.

Permissible values	Value	Meaning
	1	Fish
	2	Bird
	3	Mammal
	4	Reptile
	5	Marsupial
	8	Other

To fix this, the value domain needs to include a proviso in the 'mammals' value meaning, as shown below.

Permissible values	Value	Meaning
	1	Fish
	2	Bird
	3	Mammal other than a marsupial
	4	Reptile
	5	Marsupial
	8	Other

Use of the 'Other' permissible value

When using 'Other', to ensure an exhaustive set of permissible values, using a code value that is contiguous with the last code in the permissible value sequence should be avoided.

The following set of permissible values contains a code for 'other' that is contiguous with the other enumerated values:

Permissible values	Value	Meaning
	1	Fish
	2	Bird
	3	Mammal
	4	Reptile
	5	**Other**

This allows another enumerated category to be added to the list of permissible values without renumbering the codes or assigning a code that appears out of place in the list.

Otherwise, this would lead to a set of values as follows:

Permissible values	Value	Meaning
	1	Fish
	2	Bird
	3	Mammal
	4	Reptile
	5	**Other**
	6	Plant

or

Value	Meaning
1	Fish
2	Bird
3	Mammal
4	Reptile
5	Plant
6	**Other**

In a longitudinal data collection, renumbering of codes can be problematic as this would mean that the data for the new collection period would be different from previous periods.

If the original set of values was

Permissible values	Value	Meaning
	1	Fish
	2	Bird
	3	Mammal
	4	Reptile
	8	**Other**

it could have been easily changed to:

Permissible values	Value	Meaning
	1	Fish
	2	Bird
	3	Mammal
	4	Reptile
	5	Plant
	8	**Other**

This leaves more space for future expansion if desired.

- Avoid the use of a value for 'Other' that is commonly used as a supplementary value. This means avoiding the use of 97, 98, 99, 997, 998, 999, 9997, 9998, 9999 for the coded value of 'Other'.

Use of Supplementary values

- Data that are collected for statistical purposes needs to be precisely defined and enumerated to provide meaningful statistical information. To this end, it is important that 'missing information' be understood for statistical purposes. This means that such data will nearly always need a supplementary value to capture missing values in order to accommodate statistical analysis. To limit variations in the meaning within a specific data collection, a default supplementary value meaning, such as "Not stated/inadequately described" is used.

  For a single-character field, where possible, use:

  9 Not stated/inadequately described

  For a two-character field, where possible, use:

  99 Not stated/inadequately described

  For a three-character field, where possible, use:

  999 Not stated/inadequately described

  and so on.

- When using more than one supplementary value, use a logical set. This ensures consistent use of supplementary code values and their meanings in a value domain.

  Examples

7 (or 97, or 997 and so on)	Not applicable
8 (or 98, or 998 and so on)	Unknown
9 (or 99, or 999 and so on)	Not stated/inadequately described

  If the three supplementary values shown above are commonly used in a data set, the same code/value meaning pairs should be used in every data element. If a Supplementary value of 'Unknown' is required, but a supplementary value for 'not stated' is not required, the Code/Value meaning pair '98 Unknown' should be used in a two-character field. The entry '99 Unknown' should be avoided. This would mean that in a data collection 'unknown' would never have the coded value of 99 in one data element and 98 in another.

- Data collected that is not of statistical significance does not need a supplementary value. However, there is an exception to this rule: if the collection owners want to know the reasons for missing values (such as 'data provider refused to provide the information' or 'data provider was unable to obtain the information at this time' or 'data provider did not know the information'), then supplementary values are appropriate as possible cues for further action.

- The value 'Other' is a 'Permissible value' and is not a 'Supplementary value'. Any value that is a synonym of 'Other' (such as 'Living thing not elsewhere classified' or 'Living thing nec') is also a 'Permissible value' and is not a supplementary value.

- Just as with the use of the 'Other' permissible value, the use of values that are contiguous with the last code in the permissible value sequence should be avoided. This allows another enumerated category to be added to the list of permissible values without renumbering the codes or assigning a code that appears out of place in the list.

- It is preferable that the supplementary value field size is the same number of characters as the other permissible values in the value domain. However, this may sometimes not be possible. A set of permissible values may use up all of the codes for its specified field length, for example a single numeric field length file may use all coded values from 1 to 9. In such cases, if supplementary values are required, the field size of the data element must be increased to 2 to accommodate them

- In non-enumerated value domains (that is those without defined value meanings, such as in a measurement) the supplementary value should be one that is not possible to achieve based on the definition of a value domain. If a data element collects the measurement of the height of a person in centimetres, it would be permissible to use 997, 998 and 999 as supplementary values because the probability of a person being that tall approaches zero. It would not be valid to use a supplementary value of 99 as this could be an actual value collected.

4.3.6 Guide for use—guidelines

Guide-for-use information is intended to provide advice or interpretation on how to use particular data standards.

The Guide for Use should include information about what is collected or recorded, rather than information about how the data are collected (or recorded). Information that can be included in the Guide for Use includes:

- The meaning or interpretation of values or codes; for example, 'The start date of treatment is recorded regardless of whether treatment is completed as intended or not'.

  Or

CODE 3	Residential aged care service Includes nursing home beds in acute care hospitals.

- What is collected or included; for example 'Collected for radiation therapy and systemic therapy only. Date of surgical treatment is collected as a separate item'.
- What is excluded or not collected; for example, 'Does not include services provided through community health settings, such as community and child health centres'.
- Number of values to be collected; for example, 'More than one value can be recorded' or 'Each surgical procedure used should be recorded'.

4.3.7 Collection methods – guidelines

Collection methods include information about how data are to be collected. As such the collection methods attribute is only relevant to data elements. They do not apply to the object class, property, data element concept or value domain. Exceptions may be made for classifications and data set specifications.

In the collection methods attribute, include information about how the data are to collected or recorded, rather than information about what is collected, and the interpretation of codes or values. Information that may be included in the collection methods includes:

- Instruments or tools used; for example, 'data are obtained by asking the following question …' or 'data compiled from service records'.
- Who collects the data; for example, 'measurement of lipid levels should be carried out by laboratories which have been accredited by the National Association of Testing authorities'.
- Period for which the data are collated and reported; for example 'financial year ending 30 June each year'.
- How to record; for example 'The full name of the agent should be recorded if the coding manual is not available'.
- Other related data that are collected in conjunction; for example, 'if codes 1 or 2 are recorded, the dose of radiation received should also be recorded'. Or 'collected in conjunction with triage time'.

Recommended questions to ask when collecting data can also be included in the collection methods attribute. Standard questions, such as those recommended by the ABS, should be used where possible. If these are not available and questions have to be developed, the data developer should bear in mind that the wording of a question can result in different responses. Leading questions that give the impression that there is a correct response should be avoided. Asking two questions within the one question should be avoided. Questions should be kept short and simple, using language that is easily understood.