

Asset Management Data Standard

Data for better decisions and a better future.

AMDS Technical Specifications and Design Schema Handbook

Asset Management Data Standard and Technical Specifications for Land
Transport

WAKA KOTAHI NZ TRANSPORT AGENCY

31 JULY 2020

VERSION: 0.3

Asset Management Data Standard (AMDS)

Document History

Version	Date of Approval	Authorised by	Change Summary
v. 0.2			Consultation Draft
V0.3			

Copyright information

Copyright ©. This copyright work is licensed under the Creative Commons Attribution 4.0 International licence. You are free to copy, distribute and adapt the work, as long as you attribute the work to REG/Waka Kotahi and abide by the other licence terms. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0>

Disclaimer

REG/Waka Kotahi has endeavoured to ensure material in this document is technically accurate and reflects legal requirements. However, the document does not override governing legislation. REG/Waka Kotahi does not accept liability for any consequences arising from the use of this document. If the user of this document is unsure whether the material is correct, they should refer directly to the relevant legislation and contact REG/Waka Kotahi.

More information

If you have further queries, email us at: amds@nzta.govt.nz

The Standard and accompanying Technical Specifications can be viewed at: www.nzta.govt.nz/amds

For a limited time, interested parties can register to comment.

Table of Contents

PART 2: Technical Specifications & Schema _____ **5**

1. Technical Specifications Design _____ **5**

 1.1. Key Design Rules _____ 6

2. Data Object Relationships Model _____ **6**

3. Unique and Reusable (Normalised) _____ **7**

 3.1. Normalisation _____ 7

 3.2. Uniqueness _____ 7

 3.3. Reusable _____ 7

4. Technical Specifications Controls _____ **8**

 4.1. Obligation Levels _____ 8

 4.2. Data Validation Rules _____ 8

 4.3. Missing and Null Values _____ 9

 4.4. Null Value Justification _____ 9

 4.5. Quality Qualifier _____ 9

 4.6. Data Types and Units _____ 9

 4.7. Data Types _____ 10

 4.8. The International System of Units (SI) _____ 10

 4.9. ISO Standards _____ 10

5. Technical Specifications Definitions and Structure _____ **11**

 5.1. Attributes _____ 11

 5.2. Entity Definitions including examples _____ 12

 5.3. Attributes – Universal, Common and Specific _____ 15

 5.4. Common (Universal) Attributes _____ 16

 5.5. Universally Unique Identifier (UUID) _____ 16

 5.6. Common Attributes for Entities other than Asset _____ 16

 5.7. Unique Asset Identifier _____ 17

 5.8. Null Values _____ 17

 5.9. Value Types and Definitions _____ 17

 5.10. Descriptor Definitions and Requirements for Use _____ 19

 5.11. Data Type and Unit examples for the Pavement Layer _____ 27

6. Technical Specifications are Portable _____ **28**

 6.1. Exporting the Schema _____ 28

Table of Figures

Figure 1: Technical Specifications – common and specific attributes _____ 11

Figure 2: Image of Acoustic Barrier _____ 18

Table of Tables

Table 1: Entity Definition _____ 12

Table 2: Value List Example 1: Asset Types _____ 17

Table 3: Value List Example 2: Material Types _____ 18

Table 4: Common Descriptors that apply to all data objects _____ 20

Table 5: Optional Should/May descriptors for any data object _____ 20

Asset Management Data Standard (AMDS)

<i>Table 6: Obligations</i>	22
<i>Table 7: Descriptors for Entity Data Objects that require Location/Zone:</i>	22
<i>Table 8. Descriptors for Attribute Data Objects</i>	23
<i>Table 9. Descriptors available for Validation of Data Type Quantitative Data Attributes</i>	24
<i>Table 10. Descriptors available for Validation of Unit Quantitative Data Attributes</i>	25
<i>Table 11: Data Type and Unit examples for the Pavement Layer</i>	27

PART 2: Technical Specifications and Schema

This document should be read in conjunction with the 001 AMDS Standard – 31 July 2020 (Part 1).

Part 2 (this document) sets out:

- the structure of the technical specifications and schema based on the design principles and concepts explained in Part 1.
- the way the technical specifications are designed and the business rules for their implementation. It provides examples to contextualise the design requirements.

NOTE: The complete technical specifications are managed in the AMDS Management Tool and can be exported in full, or in part as specifically defined schemas.

1. Technical Specifications Design

Entities, entity types, attributes, descriptors and values as defined within Part 2 are collectively referred to as 'data objects'. This technical specification design establishes and defines:

- The definitions and business rules used in this standard for:
 - Entities
 - Entity Types
 - Attributes
 - Descriptors for attributes
 - Values
- The relationships between these data objects.
 - Descriptors for attributes, including the obligation levels for attributes and values for conformance with the standard.
 - Data validation rules and requirements for text-based and numeric values, null values and unique identifiers.
 - The way in which the technical specifications can be exported as a schema.

NOTE: For terminology used within the standard see PART 2: Technical Specifications & Schema and **Error! Reference source not found.**

NOTE: This section is not intended to be exhaustive and is intended to articulate the rules and provide context for their application. Examples are provided to illustrate the rules for clarity.

1.1. Key Design Rules

The key design rules ensure that the technical specifications are:

- Defined and structured: the data objects have clear and unambiguous definitions and are systematically described. The arrangement of data objects is process-driven, supports data capture through business rules, and assigns pre-defined relationships.
- Unique (Normalised) and Reusable: the data objects are unique, have unique identifiers, and are reused not duplicated.
- Controlled and extensible: data objects are systematically validated, and controlled through obligation rules
- Complete and portable: are fit-for-purpose, prevent missing values, and are exportable and transferrable between systems.
- Readable and useable: presented in both human and machine-readable formats. Therefore, the design will provide a machine-readable data schema for export/import, but also be presented in a human readable format for clarity – able to be reviewed and used by designers.

NOTE: The design approach is intended to align with the One Network Road Classification (ONRC) guidelines¹.

2. Data Object Relationships Model

The schema is defined, and constructed, through building a relational model based on the **Error! Reference source not found.** The combination of the relationships between the entity types, attributes, and values expressed as business rules.

Entity types can relate to multiple attributes.

Attributes are applied and expressed using value lists, with some operating parameters:

- An attribute can only relate to one value list (whether, conditional, dynamic, or fixed), i.e. one-to-one relationship.
- A value list can relate to multiple attributes. i.e. one-to-many.
- Values can relate to multiple value lists, i.e. many-to-many.

The data structure allows for multiple relationships between attributes and therefore the related values within any online tool or database. These relationships define the data structure, and include:

- Inheritance – Inherits / Inherited by
- Related to
- Applicable to.

¹ One Network Road Classification (ONRC): www.nzta.govt.nz/assets/Road-Efficiency-Group-2/docs/onrc-guidelines.pdf

3. Unique and Reusable (Normalised)

3.1. Normalisation

Wherever possible and useful, the specification should be normalised to:

- Enable logical groupings of data within the standard and therefore be more easily configured in an agency's database.
- Minimise duplicate definitions of data within the standard (re-using attributes).
- Enable modification in only one place (therefore replicated easily across both the 'written' standard and within any database).
- Allow maintenance of the definition of one piece of data in one place.
- Enable quick and efficient access and manipulation of the data, both within the standard and then within an agency's database.

For example, where an attribute is used for the same purpose across entities, that attribute should be shared, with ALL the descriptors for that attribute being the same.

3.2. Uniqueness

Each data object defined within the standard is to be unique where that data object has a single purpose.

A Data Object may also be considered unique where it is different from other similar data objects because of the way that data object is described and/or used. For example, the characteristic of uniqueness for an attribute can be established by:

- the unit of measurement e.g. Height (m), Height (mm)
- how the attribute is measured e.g. from the base, from the side
- relationship to a different value list, e.g. Pavement Material versus Barriers Material.

Each unique data object will be uniquely identified by its name and code descriptors.

3.3. Reusable

The technical specifications must present, use and re-use attributes and related descriptors so that they are readable, manageable, and useable:

- Where a data object forms a partial description of an entity, it is defined as an attribute. Examples: 'Material' and "Pavement Layer Thickness" are attributes of the asset type of Pavement - each partially describes the asset.
- Divisions of a characteristic that categorises something, but doesn't require any further data collection, are categories and form a Value List for an attribute defining the characteristic. Example: 'Underpass', 'Overpass' and 'Arch' can all be used to define a 'Tunnel structure type', but cannot be broken down further, therefore they are Values that describe the attribute.

A data object is to be labelled, categorised, and structured by the kind of data that it is; and the function that the data object is performing:

- Categorisation – qualitative data that describes a characteristic of an entity. Examples: 'Type' of asset, Type of 'Material' used.
- Measurement – quantitative data that describes the relevant metrics of an item or a state, or a dimension: Examples: Base course depth, Step count.

NOTE: These measurements can be raw data, derived or calculated/aggregated.

4. Technical Specifications Controls

4.1. Obligation Levels

This document defines the obligation levels² for the Technical specifications as:

- **MUST:** This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.
- **MUST NOT:** This phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.
- **SHOULD:** This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course. SHOULD is an Optional Obligation.
- **SHOULD NOT:** This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behaviour is acceptable or even useful, but the full implications should be understood, and the case carefully weighed before implementing any behaviour described with this label.
- **MAY:** This word, or the adjective "OPTIONAL", means that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option **MUST** be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option **MUST** be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.).
- **CONDITIONAL:** Some requirements will be "**CONDITIONAL**", i.e. they will only apply if contingent on another requirement, e.g. 'if x applies, then you **MUST** do y, or z', or 'if you **MUST** do x, then you **SHOULD** pick a value from list y'.

4.2. Data Validation Rules

All data should be validated before transfer between organisations.

² Adapted from: [Internet Engineering Task Force \(IETF\) RFC-2119: Key words for use in RFCs to Indicate Requirement Levels](#)

Validation is to be included within the attribute descriptors where the value for that attribute is not defined with a value list. For example, the attribute “Barrier height” has a minimum and maximum range and is in millimetres.

4.3. Missing and Null Values

One of the most important components of a data management strategy is the proper handling of missing values. Every data collection effort is essentially an attempt to answer a set of questions of this form:

What was the value of _____ at X time and place?

Missing values are a potential source of bias when conducting analysis of data sets. It is imperative that a method of recording that data is ‘missing’, and why, is standard across the data set, and that the method itself enables efficient transfer of the data and does not impact the analysis.

Occasionally data may be required before it is fully complete, e.g. in a pre-built design state before as built data is available (this is a proposed new road).

A Null value, in a database context, is the total absence of a value in a certain field and means that the field value is unknown. Null is not the same as a zero value for a numerical field, text field or space value. Null implies that a database field value has not been stored. Null can also be interpreted as zero, so the term ‘missing’ is used in this standard.

4.4. Null Value Justification

To meet the Data Validation Rules, all attribute data objects with value ‘0’ must be accompanied by a validation code, i.e. “Reason for Null Values”: These are:

1. Not available
2. Not applicable
3. Not captured
4. Value is zero

4.5. Quality Qualifier

Where data is defining a measurement, of e.g. an asset then such a measurement must be accompanied by a qualifier defining the level of accuracy of that measurement.

For example, “Length” could be surveyed, estimated, derived etc.

4.6. Data Types and Units

Data types and units are descriptors that specify the requirements for describing a value of an attribute where that attribute does not use a value from a set of values in a value list.

The chart below defines the data types and units applied within this standard. These descriptors define the minimum restrictions that can be placed on a value. If required, these can be revised within the technical specifications with altered restrictions.

Data types in the technical specifications are based on three control conventions, MySQL data types, SI Units and ISO 8601-1:2016 for the representation of Dates and Times

4.7. Data Types

MySQL Data Types³, include:

- Numeric Types e.g.
 - Integer types (exact value) INTEGER, INT, SMALL INT, TINYINT, MEDIUMINT, BIGINT (including negative values)
 - Fixed Point Types (Exact Value) DECIMAL, NUMERIC–
 - Bit-Value Type - BIT
- Date and Time Types e.g. DATE, DATETIME, TIMESTAMP, TIME, YEAR
NOTE: use of Date and Time in MySQL is an application of ISO 8601-1:2016)
- String Types e.g. CHAR, VARCHAR, BLOB, TEXT
- Spatial Data Types as per the Open Geospatial Consortium (OGC)⁴

4.8. The International System of Units (SI)

The International System of Units⁵ (SI, abbreviated from the French *Système international (d'unités)*) is the modern form of the metric system, and is the most widely used system of measurement. It comprises a coherent system of units of measurement built on

- 7 base units, which are the second, metre, kilogram, ampere, kelvin, mole, candela
- 20 prefixes to the unit names and unit symbols that may be used when specifying multiples and fractions of the units.
SI Prefixes include kilo, centi, milli, and micro
- Non-SI units accepted for use with SI Units include
- Time: minute, hour, day, (Uses ISO 8601-1:2016)
- Area: hectare;
- Volume: litre

4.9. ISO Standards

ISO 8601-1: 2016 Representation of dates and times. This establishes a standard for the interchange of date and time data in the AMDS.

- Date is represented as – YYYY-MM-DD, e.g. 2019-06-19.
- Time of day is represented as in the 24-hour clock system in the format [hh]:[mm]:[ss], e.g. 16:30:00.
- Further extensions of date and time can be accommodated in the standard so long as they conform to ISO 8601-1:2016.

³ MySQL 8.0 Reference Manual Chapter 11 Data Types <https://dev.mysql.com/doc/refman/8.0/en/data-types.html>

⁴ www.opengeospatial.org

⁵ ISO 80000-1:2009 Quantities and units – Part 1: General (includes SI units)

5. Technical Specifications Definitions and Structure

The technical specifications are defined and structured based on the AMDS Entity Model. (Refer to AMDS **Error! Reference source not found.**)

- Each entity is expressed through a controlled list of entity types.
- Entity types have controlled lists of attributes attributed to them.
- Each attribute expresses properties or characteristics of entities and entity types.

5.1. Attributes

As illustrated below, an attribute is either:

- A 'specific attribute', which is an attribute applicable to only one entity, or entity type (one-to-one),
- A 'common attribute', which can be assigned to multiple entities (one-to-many).
- Some common attributes may be universal, and applicable to all entities and entity types (one-to-all).

Technical specifications - common and specific attributes

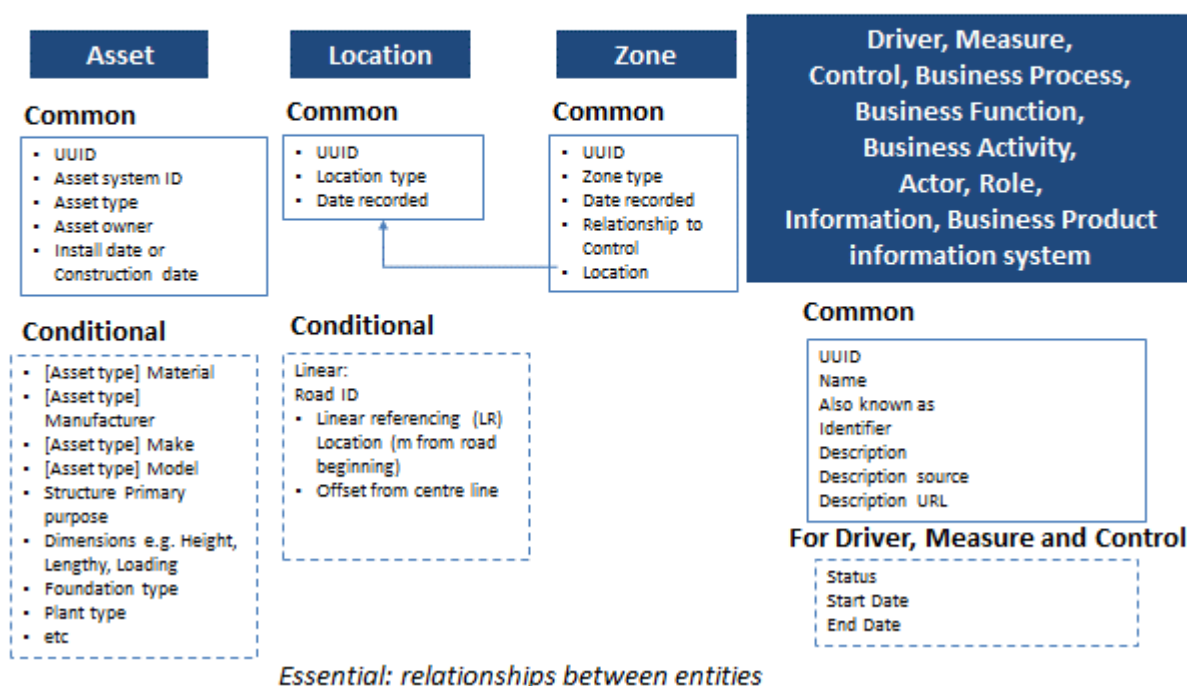


Figure 1: Technical Specifications – common and specific attributes

5.2. Entity Definitions including examples

Entities (listed and defined in **Error! Reference source not found.**, above) are essentially a high-level categorisation for grouping things that are of the same type together. These entities or groupings of ‘definable things’ are categorisations of objects, activities, places, organisations/people, etc. The use of entities enables the grouping of entity types which can have attributes applied as metadata.

The following table expands the **Error! Reference source not found.** in **Error! Reference source not found.** above, providing examples of Entity Types and relationships between entities; i.e. Asset Entity is related to Location (all assets have locations), and Zone (an Asset may be within a Zone).

Table 1: Entity Definition

Entity	Definition	Example Entity types in the Standard	Example relationships between entities
Asset	The asset is described at the “Managed Asset” level, i.e. The level at which an asset is managed in Asset Management Systems. This class contains the physical details of the asset; initially “as-built” or “as purchased/received (if vested).	<ul style="list-style-type: none"> Barrier Gantry Pole Bridge Pavement layer Surface layer Non-electronic sign Wetland 	<ul style="list-style-type: none"> Asset <i>has</i> Location is Asset <i>maintained by</i> Business Process
Location	Location refers to the ‘actual’ physical location of an asset in time and space, using linear or geospatial referencing.	<ul style="list-style-type: none"> Linear Referencing Spatial Referencing Coordinate Location 	<ul style="list-style-type: none"> Asset <i>locates</i> asset Location <i>is location of</i> Asset
Zone	<p>A three-dimensional extent or measurement of land or surface, or part of a region, that represents a virtual or conceptual view of asset(s) or a view of activities related to aggregations of assets.</p> <p>For example: how a network of linear assets is arranged conceptually for multi-asset representations, expressed as the geometry of the asset, graphically as a point, line, polygon, or other defined area, and/or a segmentation or virtual area expressed as geometry, or graphically as a point, line, polygon, or other defined area, and/or referenced using linear or geospatial referencing.</p>	<ul style="list-style-type: none"> Jurisdiction Contract zone Observation zone Road corridor zone Traffic control zone Transport usage zone 	<ul style="list-style-type: none"> Central Otago SH Network zone <i>realises</i> NOC Zone

Asset Management Data Standard (AMDS)

Entity	Definition	Example Entity types in the Standard	Example relationships between entities
	<i>NOTE:</i> Zones may also contain sub-zones.		
Business Function	<p>A high-level business function performed by, or assigned to an organisation, or its employees. This is usually defined through an enabling charter, legislative mandate, or commercial mission statement.</p> <p>Note: see also Business activity for lower levels of what Waka Kotahi does</p>	<p>Waka Kotahi Business function:</p> <ul style="list-style-type: none"> • Land transport planning and investment <ul style="list-style-type: none"> – Sector research – Planning for land transport, etc • State highway road network construction and maintenance • State highway road construction/improvements and maintenance etc 	<ul style="list-style-type: none"> • Business function is realised by Business Process • Business function is mandated by Driver
Event	A defined task that has an identifier, has a start date and end date, and runs as part of a process. Relates to a business function. Examples are Projects, a job linked to a works order for Condition Assessment, an instance of seeking a resource or building consent.	<ul style="list-style-type: none"> • Condition Assessment • Forecast/estimate • Asset defect • Outage • Reporting • Asset procurement • Asset decommissioning 	<ul style="list-style-type: none"> • Event is resolved by Business process • Event generates Information
Business Activity	Business activity has been established as a separate entity to accommodate the range of business categorisations used by Waka Kotahi. These can then be related to each other and to Business Process and Business Product, so that Business Process becomes the centre of relationships for what the business does and produces. Business activity categorisations change over time to suit business needs.	<ul style="list-style-type: none"> • NLTP work category • NOC Maintenance specification • NOC schedule of Prices. 	<ul style="list-style-type: none"> • Business Activity <i>is realised by</i> Business Process • Business Activity <i>informs</i> measure

Asset Management Data Standard (AMDS)

Entity	Definition	Example Entity types in the Standard	Example relationships between entities
Role	An actor assumes a role to perform a task. The actor in a role can be responsible for the performance of a business activity. A role can be performed within an organisation by an employee or contractor, or externally to an organisation, by a service provider. It also describes roles such as funding bodies, stakeholders, or regulators.	Administrative role: <ul style="list-style-type: none"> • Administrator • Funder • Manager • Owner • Regulator • Road controlling authority, etc Citizen Contracted Role: <ul style="list-style-type: none"> • Client • Contractor • Design engineer • Principal • Professional consultant, etc 	<ul style="list-style-type: none"> • Role <i>is performed by</i> Actor • Role <i>participates in</i> Business Process
Actor	The name of the organisation, group or individual carrying out an activity, e.g. for an individual the Person's name or organisational identity, such as an employee number/service provider reference.	<ul style="list-style-type: none"> • Transport Authority • Auckland Transport • Name of individual, e.g. Johnston, Samuel • ID of individual, e.g. NW314325D 	<ul style="list-style-type: none"> • Actor <i>resolves</i> Event • Actor <i>owns</i> Information
Control	The business rules for how an event or process must operate. Controls enable compliance activities to be enforced and monitored. A control establishes which measures may be applied in the context of a business process, such as establishing the operating parameters for managing resilience, flow, or capacity.	<ul style="list-style-type: none"> • Regulations • Speed limit • Reporting algorithm • Service level requirements • NOC • Work order 	<ul style="list-style-type: none"> • Control <i>regulates</i> Business Process • Control <i>establishes</i> Zone
Driver	A formal permission or requirement to undertake a business function or lower level activity. Includes agreement or contract- a formal document between one or more parties that sets out who will do what and conditions.	<ul style="list-style-type: none"> • Government Policy • Legislation • Government Key Result Area 	<ul style="list-style-type: none"> • Driver <i>requires</i> Measure • Driver <i>mandates</i> Business Function
Business Product	The business product is the outcome or final deliverable that is derived from a business process.	<ul style="list-style-type: none"> • Business decision • Valuation report • Condition assessment report • Building Consent • Measurement • Maintenance action 	<ul style="list-style-type: none"> • Business product <i>is delivered by</i> Business Process • Business product <i>realises</i> Business Function

Entity	Definition	Example Entity types in the Standard	Example relationships between entities
Information System	The hardware and software application(s) or set of related programs that are used to create and manage digital content.	<ul style="list-style-type: none"> Financial management System Asset Management System Enterprise Content Management System 	<ul style="list-style-type: none"> Information system <i>manages</i> Information
Business Process	A business process is a collection of related, structured activities, or tasks by actors or equipment which in a specific sequence produce a service or product which carries out a business function. It is a defined set of events that, once completed, will accomplish a predefined outcome, and sets out the 'how' a business function is undertaken, to produce a business product.	<ul style="list-style-type: none"> Barrier installation Pathway construction Pavement rehabilitation 	<ul style="list-style-type: none"> Business process <i>is performed by</i> Actor Business process <i>delivers</i> Business Product
Information	The collection of data and information which describes an asset and the activities and events that are related to it, for the purposes of asset management.	<ul style="list-style-type: none"> Rural roads service contract Defect report Condition assessment Maintenance schedule Maintenance Report and Invoice Quality assessment 	<ul style="list-style-type: none"> Information <i>is managed by</i> Information System Information <i>is generated by</i> Business Process
Measure	An indicator or factor that can be tracked, usually on an ongoing basis, to determine success or alignment with objectives and goals, e.g. performance measures such as Network performance and capability - Traffic – throughput, Travel Time	<ul style="list-style-type: none"> Investment performance Operational performance 	<ul style="list-style-type: none"> Business Activity <i>informs</i> Measure

NOTE: This table is illustrative and not intended to be exhaustive.

NOTE: An Entity is identified by being significantly different enough from any other high-level grouping of data objects as to qualify as a unique “thing”.

NOTE: Each Entity has attributes which are expressed as values and fields in asset management systems.

5.3. Attributes – Universal, Common and Specific

An attribute is a data object which describes a property or characteristic of an entity, or entity type or component.

An attribute may be denoted as a 'specific attribute', which is an attribute applicable to only one entity, or entity type (one-to-one), e.g. Asset Type or a 'common attribute', which can be assigned to multiple entities (one-to-many) e.g. Mount Type. Some common attributes may be universal, and applicable to all entities and entity types.

The diagram below shows the application of common, universal, and specific attributes applied within the context of Poles, Gantries Masts and Mounts, Brackets, and Devices.

5.4. Common (Universal) Attributes

The only common attribute is a UUID – Unique Asset Identifier. Assets have asset-specific attributes. Non-asset entity types have common attributes – see Figure 1 and 5.5 Common Attribute

5.5. Universally Unique Identifier (UUID)

A Universally unique identifier (UUID) is a 128-bit number used to identify information in computer systems. The term globally unique identifier (GUID) is also used, typically in software created by Microsoft. One of the reasons for using UUIDS is that they enable the creation of a unique ID external to a system generated ID, and also enable assignment of an ID before an internal process finishes.

All information objects created and transferred using this Standard must have a Unique ID.

How a UUID is created is not mandated. Any Online UUID generators that meet the Open Software Foundation (OSF) standards are acceptable.⁶

5.6. Common Attributes for Entities other than Asset

As in Figure 1, the following Entity types:

- Driver
- Measure
- Control
- Business Process
- Business Function
- Business Activity
- Actor
- Role
- Information
- Business System
- Business Product

have the following attributes:

- Name
- Also known as
- Identifier
- Description

⁶ (Refer ISO/IEC 11578:1996 "Information technology – Open Systems Interconnection – Remote Procedure Call (RPC)" and ITU-T Rec. X.667 | ISO/IEC 9834-8:2005.)

Asset Management Data Standard (AMDS)

- Description source
- Description URL

Drivers, measures and controls also have the following attributes:

- o Status
- o Start Date
- o End Date

5.7. Unique Asset Identifier

All New Zealand infrastructure and environmental assets must have a unique asset identifier. This is typically assigned when an asset is registered into an Asset Management System.

Before an asset has been registered within an Asset Management System, it will only have a Universally Unique Identifier (UUID).

5.8. Null Values

Null Values is a universal Attribute, because any field with a Null Value must be justified. See *Missing and Null Values*, above.

5.9. Value Types and Definitions

The Controlled Value Lists are used to populate and describe the entity types and related attributes through business rules:

Values are expressed in one of two ways:

- Value lists as text-based labels to populate descriptors. In most cases values must be derived from an agreed pre-populated value list, e.g. Asset Type list, Material list, etc.
- Numeric characters entered manually from observations, derived from calculations, or automatically captured by business systems or applications. See Data Types and Units.

NOTE: The full set of value sets are stored within a 'Master' data dictionary, within the AMDS Management Tool, and are aggregated into controlled 'pick-lists' for attribution to entity types as attributes.

The following table illustrates Controlled Value Lists for Asset Types. Each list is only used once.

Table 2: Value List Example 1: Asset Types

Example Asset Type	Definition (of asset type)	Example Values in Value List
Barrier	Physical structure used to separate modes of use or to improve safety or to block sound etc.	-vlBarrierPurpose
High mast	Pole structure that is over 16 metres high	-vlStructureMaterial
Surface layer	The surface layer is placed above the pavement layers to provide an even running surface, improved durability of the pavement and increased resistance to skidding	-vlAdhesionAgent
Non-electronic sign	Physical structure that signals a direction	-vlNonElectronicSignFunction

Asset Management Data Standard (AMDS)

NOTE: Asset Types will be captured and aggregated into the standard as they are agreed and standardised.

Material is defined here as “the matter from which a thing can be made or constructed” anything that serves as crude or raw matter to be developed.

Table 3: Value List Example 2: Material Types

Example Materials	Barrier panel material	Gantry material	Non-electronic sign substrate material
Acrylic	Y		
Aluminium	Y	-	Y
Concrete	Y	-	-
Earth	Y	-	-
Fibreglass	-	-	-
Galvanised steel	-	Y	
Plastic	Y	-	Y
Steel	Y	-	Y
Stone	Y	-	-
Timber	Y	-	Y

The Controlled Value Lists are used to populate and describe the entity types and related attributes through business rules: The following diagram illustrates the definition of a Barrier, identifying attributes with Controlled value lists and those completed using Data types and units.



Figure 2: Image of Acoustic Barrier

5.10. Descriptor Definitions and Requirements for Use

Each data object is defined by a set of descriptors, as illustrated in the following tables. These tables provide a label, an explanation, and an example of completion, plus identification of:

- Requirement for completion of the descriptor, and
- If the descriptor can be completed more than once.

Examples:

- All data objects must have one Natural Language Label which is not repeatable.
- All data objects could be 'Also known as' something else and may have more than one alias (repeatable).

NOTE: The order within these tables does not necessarily represent the display order.

Table 4: Common Descriptors that apply to all data objects

These descriptors apply to all types of data objects including entities, entity types, attributes and values lists.

Descriptors	Explanation and rules for use	Obligation	Examples	Repeatable
Natural Language Label (Name)	The name of the data object in an easily human understandable label	Must	<ul style="list-style-type: none"> Unique Asset System Identifier Vertical Clearance 	No
Code (Pascal Case)	The name of the data object in a machine-readable label, using Pascal Case (no spaces, between words, all words capitalised)	Must	<ul style="list-style-type: none"> UniqueAssetSystemID 	No
Definition (Entity and Entity type, Attribute) Description for Use – (Value List, Value)	<p>Plain English definition of the data object defining for example:</p> <p>What makes this a unique attribute? What is it and / or how it is to be used within the standard schema?</p> <p>What makes this a unique value? When or how this Value should be used.</p>	Must	For attribute 'Vertical Clearance': "Establishes minimum clearance for a structure that extends over the road shoulder. Used for calculating whether a large vehicle or load can safely pass under it"	No
Example	<p>Provides examples of the attribute and assists with determining if:</p> <ul style="list-style-type: none"> This attribute is similar to another. There is actually more than one attribute. What the 'types' of this attribute are. <p><i>NOTE:</i> Examples will demonstrate the attribute visually, or</p> <ul style="list-style-type: none"> Inform the related Value List, or Illustrate how the attribute is completed. 	Must	RAMM 12345	Yes

Table 5: Optional Should/May descriptors for any data object

The following common descriptors provide clarity but are not mandatory:

Asset Management Data Standard (AMDS)

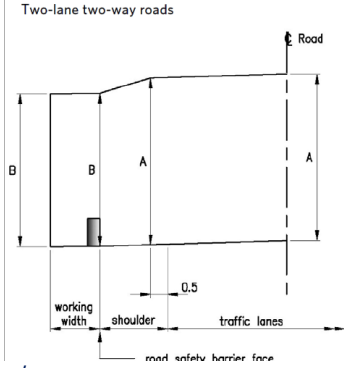
Descriptors	Explanation and rules for use	Obligation	Examples	Repeatable
Image	<ul style="list-style-type: none"> • Images of example/s provide clarity • Provide details of the image – file name, where located. • Illustration source: • NZTA Bridge Manual S3 	Should	 <p>The diagram shows a cross-section of a two-lane two-way road. It includes labels for 'working width', 'shoulder', 'traffic lanes', and 'road safety barrier face'. Dimensions 'A' and 'B' are indicated with vertical lines. A horizontal dimension of '0.5' is shown between the shoulder and traffic lanes. The text 'Two-lane two-way roads' is at the top, and 'Road' is at the top right. Below the diagram is the text 'n/a'.</p>	Yes
Also known as	<p>Identifies other terms used for this attribute.</p> <p>To ensure each attribute is included in the Standard once by referencing all terms that different organisations use for the same data entity.</p>	Conditional If applicable	<p>“Streetlight up cast angle” also known as “Tilt angle”</p>	Yes
For Further details see...	<p>For links to related websites providing additional details related to the item.</p>	Conditional If applicable	<p>ISO 8601 Representation of Dates and Times</p>	Yes

Table 6: Obligations

These additional descriptors are available for **Entity and Entity Type** data objects.

NOTE: Some entity data objects e.g. Asset and Zone will also include descriptors for Geometry/Shape and Location referencing.

NOTE: The Location parts of this standard are still under development and subject to change.

Descriptors	Explanation and rules for use	Obligation	Examples	Repeatable
Entity	To define the Entity that each specific Entity type is classified as.	Must	<ul style="list-style-type: none"> • Asset • Event 	No

Table 7: Descriptors for Entity Data Objects that require Location/Zone:

Descriptors	Explanation and rules for use	Obligation	Examples	Repeatable
Geometry /Shape	How the asset entity (asset, Business Function, etc) is represented in a graphical display.	Must	<ul style="list-style-type: none"> • Polyline • Polygon • Point <p>NOTE: Currently this equals Valid Location Type.</p>	Yes
Location reference	How the location of the entity (asset, Business Function, etc) is described.	Must	<ul style="list-style-type: none"> • Linear • Spatial <ul style="list-style-type: none"> ○ Co-ordinate ○ Well Known format 	Yes

Table 8. Descriptors for Attribute Data Objects

Additional descriptors available for any **Attributes**

Descriptors	Explanation and rules for use	Obligation	Examples	Repeatable
Attribute – Obligation	<p>The obligation level for the data object must be indicated as:</p> <ul style="list-style-type: none"> • Must (Mandatory) • Conditional • Should or May (Optional) 	Must	For Vertical Clearance: “Conditionally”	No
Obligation condition	Free text definition of the condition ⁷	Condition	“Use when the asset extends above the road or roadside”	No
Applicability to Entities and/or Entity type(s)	Identifies which entities and entity types use this attribute	Must	Vertical Clearance is Mandatory Attribute for Bridge, Pole, Gantry, Overhead wire, etc.	Yes
Data Type	That specifies the type of data that the object can hold: a mix of character data (e.g. VARCHAR) numeric integer data, monetary data, date and time data, binary strings, and so on. Data types are based on MySQL data types.	Must	VARCHAR(N), INT (N) etc	No
Value List	Identifies the values that are available for this attribute. Typically used for character data.	Conditional Complete if Value List	For Pole, use Pole Material list.	No
Business Rule	Provides informative text for a business rules on how to populate the attribute.	Conditional If there is a dependency between two attributes.	End Node must be larger than Start Node.	Yes

⁷ Note: In future development of the AMDS more conditions may be identified that could be triggered by a value (number or text) in another attribute.

Table 9. Descriptors available for Validation of *Data Type* Quantitative Data Attributes

Note: Use in conjunction with *Data Type* (above)

Descriptors	Explanation and rules for use	Obligation	Examples	Repeatable
Precision	The total number of digits in a number	Conditional – Must complete for a quantitative data object	For Pavement Layer e.g. 4 Examples: <ul style="list-style-type: none"> 1500 – where Precision = 4, Scale is 0 1.500 – where Precision = 4, Scale = 3 	No
Scale	<ul style="list-style-type: none"> If a decimal data type, the number of digits on the right-hand side of the decimal point. The sum of the number of digits on both sides must equal the Precision 	Conditional – Must complete for a quantitative data object	See Precision, above.	No
Min	<ul style="list-style-type: none"> The minimum value the attribute can have where it is not defined within the Data Type or is required to be more restrictive Not completed when Value List or standard Data Type values are used 	Conditional – Should complete for a quantitative data object	Pavement Layer Thickness minimum: = 0 mm	No
Max	<ul style="list-style-type: none"> The maximum value the attribute can have where it is not defined within the Data Type or is required to be more restrictive Not completed when Value List or standard Data Type values are used 	Conditional – Should complete for a quantitative data object	Pavement Layer Thickness maximum: = 99999 mm	No

Table 10. Descriptors available for Validation of *Unit Quantitative Data Attributes*

Note: Use in conjunction with *Data Type* (above)

Descriptors	Explanation and rules for use	Obligation	Examples	Repeatable
Unit measurement system	Use to identify what system of units are being used, e.g. SI, ISO 8601 or other specialised units such as MESA.	Conditional – Must complete for a quantitative data object	SI ISO	No
Base Quantity	SI Base quantities are Time, Length, Mass, Electric Current, Temperature, Amount of substance, Luminous intensity	Conditional – Must complete for a quantitative data object	“Length”	No
Derived Quantity	Quantities express the type of derived unit	Conditional – Must complete where quantity is derived	Area (m ²) calculated from multiplying one or more quantities.	No
Unit of measurement	Use to show that the measurement is in e.g. metres (or rather than feet), degrees Celsius rather than Fahrenheit or Kelvin etc.	Conditional – Must complete for a quantitative data object	“Metre” SI Unit Types include the second for time, and the metre, the kilogram for mass, ISO 8601 for DATE and TIME etc. •	No
Unit measurement symbol	Use to show the symbol for the unit	Conditional – Must complete for a quantitative data object	<ul style="list-style-type: none"> • m for metre • mm for millimetre • MPa for MegaPascal 	No
Base Unit	For example, the second and the metre.	Conditional – Must complete for a quantitative data object	Metre (for length)	No
Prefix	A unit prefix that precedes a basic unit of measurement to indicate a multiple or fraction of the unit.	Conditional – Use where data is captured in a sub-unit a base unit	SI prefixes for the base quantity of length include the centi, milli and the micro. Data may be captured in millimetres as a sub-unit of metres	No

Asset Management Data Standard (AMDS)

Descriptors	Explanation and rules for use	Obligation	Examples	Repeatable
Derived Unit	<p>SI derived units are units of measurement derived through a calculation of one or more units.</p> <p>They are either dimensionless or can be expressed as a product of one or more of the base units</p>	Conditional – Must complete for a quantitative data object where	For example, the derived unit of square metre m ² (for area) or metres/second (for velocity)	No

5.11. Data Type and Unit examples for the Pavement Layer

Table 11: Data Type and Unit examples for the Pavement Layer

Entity – Type	Attribute Name	Example Value	Data Type					Unit								
			Data Type	Precision	Scale	Min	Max	Unit Measurement system	Unit of measurement	Unit Measure Symbol	Base Quantity	Derived Quantity	Base Unit	Prefix	Derived Unit	
Asset – Pavement Layer	Pavement Layer Thickness	400 mm	INT	5	0	0	99999	SI	millimetres	mm	'length'	-	metres	milli	-	
Asset – Marking	Marking thickness	150 microns	INT	3	0	100	200	SI	micron		'length'	-	micro	-	-	
Asset – Pavement Layer	Design loading – Pavement	1.50 Mesa	DEC	4	2	-	-	Non-Standard	Million Equivalent standard axles	MESA	'mass'	'force'	-	million	ESA	
Asset – Pavement Layer	Layer CBR (%)	20	INT	4	1	0	120	SI	percent	%	'mass'	'force'	-	-	%	
Asset – Pavement Layer	Pavement Layer Strength UCS	1.2	DEC	4	1	0.7	1.5	SI	MegaPascal	MPa	-	'pressure'	pascal	mega	Pascal	
Asset – Pavement Layer	Date completed	2019-05-19	DATE	-	-	-	-	ISO	-	YYYY-mm-dd	'time'	-	day	-	-	
Asset – Pavement Layer	Time completed	16:30:00	TIME	-	-	-	-	ISO	-	hh:mm	'time'	-	second	-	-	

NOTES

- The descriptors are worked in full in the table above, BUT only those shaded in green are variables that must be populated in this Technical Specification, since the others will be inherited.
- Base Quantities source list is International System of Quantities, ISQ – see ISO 8000-1:2009 and https://en.wikipedia.org/wiki/List_of_physical_quantities
- For SI Derived Units, see https://en.wikipedia.org/wiki/SI_derived_unit
- CBR stands for Californian Bearing Ratio – for the natural ground and granular (non-modified) Pavement layer. In RAMM, Pavement Layer Strength is Layer Strength. UCS stands for Unconfined Compressive Strength.

6. Technical Specifications are Portable

6.1. Exporting the Schema

The AMDS is a live 'point-in-time' set of information entities, entity types, and attributes arranged in a schema that can be downloaded in formats that are fit-for-purpose for the user-base (e.g. xml, csv, and html) formats to provide:

- The full Schema for transformation by third party applications.
- Details of Entities and Attributes by defined exports which enables users to obtain normalised or de-normalised data to use for:
 - Analysis and comparison with Asset Management Systems.
 - Transformation by third party applications.