

Public Record Office Victoria

Specification

PROS 25/02 S5: ADDING METADATA PACKAGES TO VEOs

Version number: 1.0 Issue Date: 23/06/2025 Expiry Date: 23/06/2035

About this Specification

Digital records which have permanent value must be transferred to PROV as VERS Encapsulated Objects (VEOs), at a time agreed between PROV and the public office.

This Specification is for technical developers building systems and tools to construct VEOs. It sets out the requirements for adding metadata packages to VEOs.

The diagram below shows the relationship between this Specification and related documents.

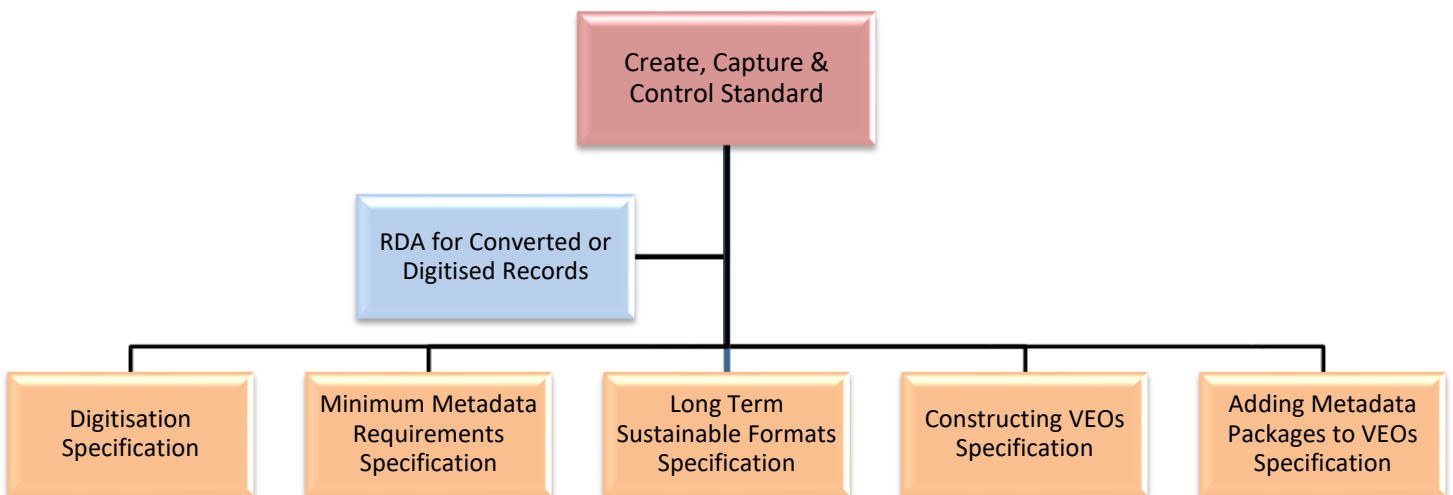


Table of Contents

Introduction	4
Authority of Standards and Specifications	4
Obligation	4
Applying this Specification	4
1 Adding Metadata to VEOs	5
1.1 Overview	5
1.2 References	5
2 Basic requirements	7
2.1 Minimum metadata in an Information Object	7
2.2 Recommended metadata requirements	7
2.3 Additional metadata packages	7
3 Additional general metadata	8
3.1 vers:UseFor element	8
3.2 vers:ContextPath element	8
4 AS/NZS 5478 Metadata	11
4.1 Standards	11
4.2 AS/NZS 5478 Metadata Package preamble	11
4.3 Basic record metadata	12
4.4 Mapping AS/NZS 5478 and RDF property names	13
4.5 AS/NZS 5478 controlled values	15
4.6 Expressing AS/NZS 5478 property values in XML	16
4.7 Expressing relationships in a VEO	16
5 General XML Metadata	20
5.1 Standards	20
5.2 XML Metadata Package preamble	20
5.3 The value of the Metadata Schema Identifier	21
5.4 Example of XML metadata	21
6 Standard aspects of RDF/XML	22
6.1 Metadata expressed as RDF	22
6.2 The value of the rdf:about attribute	22
6.3 ISO 8601 date/time values	23
6.4 XML value and attribute encoding	24
Appendix 1 AGLS Metadata (Deprecated)	25
Appendix 1.1 Standards	25

Appendix 1.2	Additional AGLS Properties	25
Appendix 1.3	AGLS Metadata Package preamble	27
Appendix 1.4	Basic record metadata	28
Appendix 1.5	Mapping AGLS Property names to XML elements	28
Appendix 1.6	AGLS controlled values	30
Appendix 1.7	Expressing AGLS property values in XML	30
Appendix 1.8	Expressing AGLS relationships	30

Introduction

Authority of Standards and Specifications

Under section 12 of the *Public Records Act 1973*, the Keeper of Public Records ('the Keeper') is responsible for the establishment of Standards for the efficient management of public records and for assisting Victorian public offices to apply those Standards to records under their control.

Heads of public offices are responsible under section 13b of the *Public Records Act 1973* for carrying out a program of efficient management of public records. The program of records management needs to cover all records created by the public office, in all formats, media and systems across the organisation.

The Standards and Specifications support the Victorian Electronic Record Strategy (VERS), which is designed to ensure the creation, capture and preservation of authentic, complete and meaningful digital records by the Victorian public sector.

This Specification is part of the PROS25/02 Create, Capture and Control Standard.

Obligation

It is mandatory for all Victorian public offices to follow the principles and comply with the requirements of the Standards issued by the Keeper. Some of the Standards have Specifications, which provide detailed technical requirements that must be complied with by Victorian public offices.

Applying this Specification

All Victorian public offices must transfer permanent value digital records to PROV as VEOs. This Specification aligns with the PROV Minimum Metadata Requirement Specification, which all Victorian public offices must meet.

This Specification sets out the requirements for adding metadata packages to VEOs. It is written for technical developers who will be building systems and tools to construct VEOs.

This Specification replaces the previous version of *PROS 19/05 S5 Adding metadata packages to VEOs* dated 27 July 2023. This version of the Specification was issued primarily to give developers clearer and more explicit technical guidance.

1 Adding Metadata to VEOs

1.1 Overview

Metadata in a VEO is contained within Metadata Packages. Every Information Object contains at least one, and often several, Metadata Packages.

Each Metadata Package is intended to contain the properties (metadata) associated with a particular metadata standard or collection. For example, if an Information Object is described using metadata drawn from descriptive, digitisation, and geospatial metadata standards, it will contain three Metadata Packages, one for each standard. Note that this means that some metadata might be duplicated, for example, each metadata standard (and hence Metadata Package) is likely to have an element representing the title.

A Metadata Package may not be associated with a standard. It may simply contain a custom collection of metadata brought together for a particular purpose. For example, the specific system metadata held by a record system about an object could be collected in a non-standard Metadata Package. Another example is where a record system provides a set of metadata elements that are presented in the user interface for users to search for records. This set of elements can be collected into a Metadata Package and labelled with its purpose. This will allow PROV to build matching functionality in our archival management system.

In this specification we explicitly state how metadata is expressed in a Metadata Package. The specification states:

- Minimum requirements on Metadata Packages within an Information Object.
- How to express two metadata standards (AS/NZA 5478 and AGLS/Dublin Core) in a Metadata Package.
- How to express non-standard metadata in generic XML.
- Additional VERS specific metadata elements that can be used in Metadata Packages.

1.2 References

This specification references the following documents:

1.2.1 Extensible Markup Language (XML)

XML is used to represent the metadata and is defined in [Extensible Markup Language \(XML\) 1.0 \(Fifth Edition\)](#).

1.2.2 Resource Description Framework (RDF)

The RDF is a framework produced by the W3C to represent information in the web (including metadata). Specifically, it defines a way of representing a graph of resources and properties (metadata) about that resource.

The framework consists of a number of recommendations. The [RDF 1.1 Concepts and Abstract Syntax](#) specifies the RDF data model. The [RDF 1.1 XML Syntax](#) specifies how this model is expressed in XML.

1.2.3 AS/NZS 5478

The Australian/New Zealand Standard [AS/NZS 5478:2015](#) - Recordkeeping Metadata Element Reference Set (RMPRS), defines a set of properties that can be used to describe records and recordkeeping systems.

AS/NZS 5478 was based on work jointly produced by the National Archives of Australia (NAA) and Archives New Zealand. We consider the NAA product [The Australian Government Recordkeeping Metadata Standard \(AGRKMS\)](#) to be equivalent to AS/NZS 5478 and may be used.

1.2.4 AS 5044 - Australian Government Locator Service (AGLS)

The AGLS Metadata Standard is a set of descriptive properties intended to improve the visibility and availability of online resources. The formal definition of AGLS is documented in the Australian Standards [AS 5044.1:2010](#), however, we consider the [version](#) formerly published by the NAA as equivalent.

Victorian Government policy is that all web resources are described using AGLS – the [Victorian Government Implementation Guide](#) has further advice.

In this specification, properties have been added to AGLS to document when disposal of the record should be determined, and to document the context of the record.

2 Basic requirements

2.1 Minimum metadata in an Information Object

Every Information Object in a VEO must contain an AS/NZS 5478 Metadata Package¹². This metadata package must be expressed in XML using RDF.

The metadata within each standard Metadata Package must conform to PROS 25/02 S2.

2.2 Recommended metadata requirements

Metadata standards typically define many metadata properties. As many of these properties as possible should be populated from information available in the source record system.

2.3 Additional metadata packages

Information Objects may contain any number of additional Metadata Packages. Each Metadata Package represents a distinct collection of metadata. These collections may be drawn from external standards, or they may be custom collections put together by vendors to meet user or vendor needs.

Ultimately, all metadata in a Metadata Package must be physically represented as XML^{3,4}.

The 'MetadataSchemaIdentifier' property is used to identify the metadata package, and the 'MetadataSyntaxIdentifier' property is used to identify the standard used to encode the metadata into XML.

Implementers are encouraged to use well known metadata packages, and to express them using RDF.

¹ The original version of this specification only required the topmost Information Object to contain a standard metadata package. VEOs produced by products certified against the original specification will be accepted even though they only contain the standard metadata packages in the topmost Information Object.

² The original version of this specification supported either AS/NZS 5478 or AGLS metadata as the standard metadata package. AGLS, however, has now been deprecated. Existing implementations of this specification may continue to use AGLS metadata.

³ The ability to include arbitrary metadata packages allows the public office to store specific metadata in the records. Examples of metadata packages could include GIS data, and digitisation data.

⁴ The ability to be able to process and provide access to the additional metadata packages in the future will vary. It is anticipated that an archive will always allow the metadata to be downloaded and processed. In addition, the raw XML can be displayed to users. If the metadata is expressed as RDF, the metadata can be displayed and it may be possible to query the RDF.

3 Additional general metadata

PROV has defined two general purpose metadata elements that can be added to any Metadata Package (including AGLS or AS/NZS 5478 Metadata Packages).

3.1 vers:UseFor element

This element is used to indicate the purpose of the metadata collection in this Metadata Package within the originating system. For example, some systems present their users with a set of metadata properties that can be used to search for objects, or a set of metadata properties that can be used to filter (or facet) search results.

Such collections of metadata can be collected into a Metadata Package and labelled with a vers:UseFor element to show their intended purpose.

The element is optional, may be repeated, and may be used in any vers:MetadataPackage element.

Use For		
OBLIGATION: Optional/Repeatable		
Sub-properties	Notes on use	Rendered
N/A	Indicates the purpose of this metadata package. The following values can be used: <ul style="list-style-type: none">• "SearchTerms" if the metadata was presented as specific search options.• "Filtering" if the metadata is presented after information is displayed to filter (or facet) search results.• Other values as required.	vers:UseFor

3.2 vers:ContextPath element

This element describes a context of the record contained within the VEO. Typical contexts are business classification schemes, computer file systems, and URIs. Records may have multiple contexts, and hence multiple Context Path properties.

The vers:ContextPath element is particularly useful for document centric record systems. These systems register individual documents and do not have a fixed structure in which the documents are placed. Instead, each document is labelled (these labels can be structured hierarchically). The labels are used to build a dynamic structure. These labels can be placed in the vers:ContextPath element.

A Context Path is only defined for the record (VEO) as a whole and hence should only appear in Metadata Packages in the root Information Objects.

The element is optional, may be repeated, and may be used in any vers:MetadataPackage element.

A versterms:contextPath has two subelements: versterms:contextPathDomain; and versterms:contextPathValue.

Context Path		
OBLIGATION: Optional/Repeatable		
Sub-properties	Notes on use	Rendered
contextPathDomain contextPathValue	Indicates the context of this record (VEO) within a record system.	versterms: ContextPath

The optional versterms:contextPathDomain gives the domain of the context path (e.g. filepath, URI, URL, Sharepoint).

Context Path Domain		
OBLIGATION: Optional/Not Repeatable		
Sub-properties	Notes on use	Rendered
	<p>Indicates the domain of the Context Path value.</p> <p>The following values can be used:</p> <ul style="list-style-type: none"> • "Filepath" if the Context Path Value is a computer file path • "BCS" if the Context Path Value is drawn from a Business Classification Scheme. • "URI" if the Context Path Value is a Uniform Resource Identifier (URI). • "URL" if the Context Path Value is a Uniform Resource Locator (URL). • "SharepointURL" if the Context Path Value is a Sharepoint URL. • Other values as required. 	versterms: ContextPathDomain

The mandatory versterms:contextPathValue gives the value of the context path.

Context Path Value		
OBLIGATION: Mandatory/Not Repeatable		
Sub-properties	Notes on use	Rendered
	<p>This property contains the actual Context Path. The context is a hierarchical namespace, with the hierarchical elements separated by a forward slash (/) - as in a URL or a filepath. If a name component contains a '/' character, that character must be replaced by a "%2F" (note that this is the conventional encoding mechanism used for '/' in name components in a URL).</p>	<p>versterms: ContextPathValue</p>

4 AS/NZS 5478 Metadata

This section states how a metadata package containing AS/NZS 5478 metadata is to be represented in a VEO.

4.1 Standards

Information Objects may be described using metadata conformant to AS/NZS 5478.

Metadata compliant with the [Australian Government Recordkeeping Metadata Standard Version 2.2 \(AGRkMS\)](#) (June 2015) is considered to be conformant to AS/NZS 5478.

4.2 AS/NZS 5478 Metadata Package preamble

Every Metadata Package containing AS/NZS 5478 metadata must commence and end as follows:

```
<vers:MetadataPackage xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <vers:MetadataSchemaIdentifier>
    http://www.prov.vic.gov.au/vers/schema/ANZS54785
  </vers:MetadataSchemaIdentifier>
  <vers:MetadataSyntaxIdentifier>
    http://www.w3.org/1999/02/22-rdf-syntax-ns#6
  </vers:MetadataSyntaxIdentifier>
  <rdf:RDF xmlns:anzs5478="http://www.prov.vic.gov.au/vers/schema/ANZS5478">
    <rdf:Description rdf:about="[Record Entity Identifier]">
      <anzs5478:Record rdf:parseType="Resource">
        [Record metadata...]
      </anzs5478:Record>
    </rdf:Description>
  </rdf:RDF>
</vers:MetadataPackage>
```

Notes:

- The indentation is optional but makes the XML easier to read. If indentation is used, spaces are preferred as they do not take up as much horizontal space.
- The xmlns:rdf namespace is defined as an attribute in the vers:MetadataPackage element.
- The value of the vers:MetadataSchemaIdentifier element is 'http://www.prov.vic.gov.au/vers/schema/ANZS5478'. This URL is an identifier and does not point to a web page.
- The value of the vers:MetadataSyntaxIdentifier is 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'.
- The xmlns:anzs5478 namespace is defined as an attribute in the rdf:RDF element.
- The rdf:about attribute in the rdfDescription element contains a URL that uniquely identifies this record. The value for this identifier is described in section 6.2.

⁵ This URL does need not link to a web page

⁶ This URL does need not link to a web page

4.3 Basic record metadata

Each AS/NZS 5478 metadata package contains exactly one AS/NZS5478 Record entity. Mapped to XML, a Record entity with the bare minimum of metadata has the following form. Text in **bold** is replaced by values from your system.

```
<anzs5478:Record rdf:parseType="Resource">
  <anzs5478:EntityType rdf:datatype="xs:string">Record</anzs5478:EntityType>
  <anzs5478:Category rdf:datatype="xs:string">Item</anzs5478:Category>
  <anzs5478:Identifier rdf:parseType="Resource">
    <anzs5478:IdentifierString rdf:datatype="xs:string">
      [Your system identifier for the Record entity]
    </anzs5478:IdentifierString>
  </anzs5478:Identifier>
  <anzs5478:Name rdf:parseType="Resource">
    <anzs5478:NameWords rdf:datatype="xs:string">
      [The title of the record entity]
    </anzs5478:NameWords>
  </anzs5478:Name>
  <anzs5478:DateRange rdf:parseType="Resource">
    <anzs5478:StartDate rdf:datatype="xs:datetime">
      [The date/time the record entity was created in ISO 8601 form]
    </anzs5478:StartDate>
  </anzs5478:DateRange>
  <anzs5478:Disposal rdf:parseType="Resource">
    <anzs5478:RetentionAndDisposalAuthority rdf:datatype="xs:string">
      [The RDA or GDA governing transfer to PROV of this record]
    </anzs5478:RetentionAndDisposalAuthority>
    <anzs5478:DisposalClassID rdf:datatype="xs:string">
      [The clause within the RDA or GDA governing transfer to PROV of this record]
    </anzs5478:DisposalClassID>
    <anzs5478:DisposalAction rdf:datatype="xs:string">
      [The disposal action as planned in the RDA or GDA]
    </anzs5478:DisposalAction>
  </anzs5478:Disposal>
</anzs5478:Record>
```

Notes:

- It is expected that the anzs5478:Record generated from a real record system will contain much more metadata than is shown here.
- All XML elements that have subelements (e.g. anzs5478:Record, anzs5478:Identifier) have an rdf:parseType attribute with the value 'Resource'. In the table in section 4.4 these are labelled in the third column with the RDF Data Type of 'Container'
- All XML elements that DO NOT have subelements (e.g. anzs5478:EntityType, anzs5478:IdentifierString) have an rdf:datatype element that contains the data type of the element value. This is normally 'xs:string', but in some cases will be different. The actual data type to be used for each AN/NZS 5478 property is shown in the third column in the table in section 4.4.
- The value of the anzs5478:Identifier element will often be included in the rdf:about attribute (see section 6.2)

- The value of the anz5478:StartDate is an ISO 8601 date/time. The exact format of an ISO 8601 date/time is described in section 6.3.
- The value of anz5478:RetentionAndDisposalAuthority is the title of the PROV Retention and Disposal Authority (RDA), General Disposal Authority (GDA), or Single Instance Disposal Authority (SIDA) that governs disposal of records of this type. If there is no RDA, GDA, or SIDA covering the disposal of the records, the value 'No Disposal Coverage' should be used; in this case the anz5478: DisposalClassID and anz5478:DisposalAction properties will not be present.
- The value of anz5478:DisposalClassID is the class within the RDA or GDA that governs the disposal.
- The value of anz5478:DisposalAction is what the RDA or GDA states what should happen to the record when disposal occurs. For records transferred to PROV, the value would normally be 'Retain as State Archives'.

4.4 Mapping AS/NZS 5478 and RDF property names

The following mapping between AS/NZS 5478 or AGRkMS properties and RDF property names must be used. The AS/NZS 5478 or AGRkMS documents should be consulted for the meanings of the properties, and whether the properties must or may be present, and whether they can be repeated.

ASNZS 5478 Property		RDF Property (XML Element) Name	RDF Data Type
0	Entity Type	azns5478:EntityType	xs:string
1	Category	azns5478:Category	xs:string
2	Identifier	azns5478:Identifier	Container
2.1	Identifier String	azns5478:IdentifierString	xs:string
2.2	Identifier Scheme	azns5478:IdentifierScheme	xs:string
3	Name	azns5478:Name	Container
3.1	Name Words	azns5478:NameWords	xs:string
3.2	Name Scheme	azns5478:NameScheme	xs:string
4	Date Range	azns5478:DateRange	Container
4.1	Start Date	azns5478:StartDate	Note 1
4.2	End Date	azns5478:EndDate	Note 1
5	Description	azns5478:Description	xs:string
6	Related entity	azns5478:RelatedEntity	Container
6.1	Assigned entity ID	azns5478:AssignedEntityID	xs:string
6.2	Assigned entity ID scheme	azns5478:AssignedEntityIDScheme	xs:string
6.3	Relationship role	azns5478:RelationshipRole	xs:string
7	Change History	azns5478:ChangeHistory	Container
7.1	Property name	azns5478:PropertyName	xs:string
7.2	Prior Value	azns5478:PriorValue	xs:string
7.3	Relationship ID	azns5478:RelationshipID	xs:string
8	Jurisdiction	azns5478:Jurisdiction	xs:string

ASNZS 5478 Property		RDF Property (XML Element) Name	RDF Data Type
9	Security classification	azns5478:SecurityClassification	xs:string
10	Security caveat	azns5478:SecurityCaveat	Container
10.1	Caveat text	azns5478:CaveatText	xs:string
10.2	Caveat category	azns5478:CaveatCategory	xs:string
11	Permissions	azns5478:Permissions	Container
11.1	Permission text	azns5478:PermissionText	xs:string
11.2	Permission type	azns5478:PermissionType	xs:string
12	Rights	azns5478:Rights	Container
12.1	Rights statement	azns5478:RightsStatement	xs:string
12.2	Rights type	azns5478:RightsType	xs:string
12.3	Rights status	azns5478:RightsStatus	xs:string
13	Contact	azns5478:Contact	Container
13.1	Contact details	azns5478:ContactDetails	xs:string
13.2	Contact type	azns5478:ContactType	xs:string
14	Position	azns5478:Position	xs:string
15	Language	azns5478:Language	xs:string
16	Coverage	azns5478:Coverage	Container
16.1	Jurisdictional coverage	azns5478:JurisdictionalCoverage	xs:string
16.2	Temporal coverage	azns5478:TemporalCoverage	xs:string
16.3	Spatial coverage	azns5478:SpatialCoverage	xs:string
17	Keyword	azns5478:Keyword	Container
17.1	Keyword term	azns5478:KeywordTerm	xs:string
17.2	Keyword ID	azns5478:KeywordID	xs:string
17.3	Keyword Scheme	azns5478:KeywordScheme	xs:string
17.4	Keyword Scheme Type	azns5478:KeywordSchemeType	xs:string
18	Disposal	azns5478:Disposal	Container
18.1	Retention and Disposal Authority ⁷	azns5478:RetentionAndDisposalAuthority	xs:string
18.2	Disposal Class ID	azns5478:DisposalClassID	xs:string
18.3	Disposal Action	azns5478:DisposalAction	xs:string
18.4	Disposal Trigger Date	azns5478:DisposalTriggerDate	xs:string
18.5	Disposal Action Due	azns5478:DisposalActionDue	xs:string
19	Format	azns5478:Format	Container
19.1	Format Name	azns5478:FormatName	xs:string
19.2	Format Version	azns5478:FormatVersion	xs:string
19.3	Creating Application Name	azns5478:CreatingApplicationName	xs:string
19.4	Creating Application Version	azns5478:CreatingApplicationVersion	xs:string

⁷ This property is named 'Records Authority' in the AGRKMS.

ASNZS 5478 Property		RDF Property (XML Element) Name	RDF Data Type
19.5	Format Registry	azns5478:FormatRegistry	xs:string
19.6	Format Registry ID	azns5478:FormatRegistryID	xs:string
20	Extent	azns5478:Extent	Container
20.1	Physical Dimensions	azns5478:PhysicalDimensions	xs:string
20.2	Logical Size	azns5478:LogicalSize	xs:string
20.3	Quantity	azns5478:Quantity	xs:string
20.4	Units	azns5478:Units	xs:string
21	Medium	azns5478:Medium	xs:string
22	Integrity Check	azns5478:IntegrityCheck	Container
22.1	Hash Function Name	azns5478:HashFunctionName	xs:string
22.2	Message Digest	azns5478:MessageDigest	xs:string
23	Location	azns5478:Location	xs:string
24	Document Form	azns5478:DocumentForm	xs:string
25	Precedence	azns5478:Precedence	xs:string

Note 1: The RDF Data Type to be used for an azns5478:StartDate or azns5478:EndDate is described in section 6.3.

4.5 AS/NZS 5478 controlled values

A number of the properties in AS/NZS 5478 and AGRkMS have controlled values which are referred to as Encoding Schemes. In representing these values in VERS, the text used in the XML value must be identical to that presented in AS/NZS 5478 or AGRkMS with no leading or trailing spaces.

It should be noted that AGRkMS provides several additional Encoding Schemes over AS/NZS 5478. These are:

- Extensible List of Identifier Schemes (D3)
- AGLS Jurisdiction Scheme (D7)
- Protective Security Manual Security Classifications (D8)
- Protective Security Manual Security Caveat Categories (D9)
- Protective Security Manual Security Clearances (D10)
- Rights Type Scheme (D12.1)
- Rights Status Scheme (D12.2)
- Contact Type Scheme (D13)
- Getty Thesaurus of Geographic Names Online (D16)
- Extensible list of Keyword Schemes (D17)
- Digital Units Scheme (D19)
- Hash Function Scheme (D20)
- Document Form Scheme (D21)
- Document Precedence Scheme (D22)

These schemes may be used in VEOs if desired.

4.6 Expressing AS/NZS 5478 property values in XML

AS/NZS 5478 property values expressed in XML must be XML safe. The necessary encoding is described in section 6.4.

4.7 Expressing relationships in a VEO

ASNZS 5478 describes a five entity model: Record; Agent; Business; Mandate; and Relationship.

- Record entities represent the primary information captured by a record system. This includes the record itself, subsidiary parts of records, and its structural information (e.g. a business classification scheme). Every Metadata Package containing ASNZS 5478 metadata must contain exactly one Record entity.
- Agents are people, organisations, organisational units, or software that had some linkage with the record (e.g. created the record, owns it). Record systems typically include information about agents that interacted with records.
- Business entities represent a business function, activity, or transaction performed by an agent. We do not commonly see information about business entities captured in current record systems.
- Mandate entities represent a source of business requirements about records (e.g. a document requiring the creation or retention of particular records). We do not commonly see information about mandates captured in current record systems.
- Relationship entities describe relationships between the other four entities (Record, Agent, Business, and Mandates). Given that Business and Mandate entities are rarely represented in current record systems, Relationship entities normally link two Record entities or Record and Agent entities. In ASNZS 5478 relationships are timebound; they have a time the relationship starts and a time it ends (if an end date/time is not specified, the relationship is continuing).

VEOs containing ASNZS 5478 metadata packages will include details about Agents related to the record entity and (possibly) relationships to other Record entities. This section describes how these Agents and Relationships are represented in a VEO in XML.

4.7.1 Relationships with other AS/NZS5478 Record entities

If the Record entity contained in the Metadata Package is related to another Record entity in your system, this relationship must be expressed in the Record entity being generated. The relationship must be included even if the related entity is not being exported as a VEO.

A relationship with another Record entity is expressed as an `anzs5478:Relationship` contained in the `anzs5478:Record` element. A relationship has the form:

```
<anzs5478:Record rdf:parseType="Resource">
  [Other ANZS5478 metadata]
  <anzs5478:Relationship rdf:parseType="Resource">
    <anzs5478:EntityType rdf:datatype="xs:string">Relationship</anzs5478:EntityType>
    <anzs5478:Category rdf:datatype="xs:string">Provenance Relationship</anzs5478:Category>
    <anzs5478:Identifier rdf:parseType="Resource">
      <anzs5478:IdentifierString rdf:datatype="xs:string">
        [Identifier of relationship]
```

```

</anzs5478:IdentifierString>
</anzs5478:Identifier>
<anzs5478:Name rdf:parseType="Resource">
  <anzs5478:NameWords rdf:datatype="xs:string">
    [Type of Relationship]
  </anzs5478:NameWords>
</anzs5478:Name>
<anzs5478:DateRange rdf:parseType="Resource">
  <anzs5478:StartDate rdf:datatype="xs:datetime">
    [Date/time relationship was established]
  </anzs5478:StartDate>
</anzs5478:DateRange>
<anzs5478:RelatedEntity rdf:parseType="Resource">
  <anzs5478:AssignedEntityID rdf:datatype="xs:string">
    [Your system identifier of the related Record entity]
  </anzs5478:AssignedEntityID>
  <anzs5478:RelationshipRole rdf:datatype="xs:string">
    [Direction of the relationship]
  </anzs5478:RelationshipRole>
</anzs5478:RelatedEntity>
</anzs5478:Relationship>
</anzs5478:Record>

```

Notes:

- Only the metadata related to the relationship is included in this XML snippet.
- The value of the anzs5478:IdentifierString element is a unique identifier of the relationship itself. If your record systems does not allocate a unique identifier to relationships use the value of Record entity anzs5478:IdentifierString concatenated with a sequential number within the VEO. For example, if the Record entity had an anzs5478:IdentifierString with the value 'R3345', the first relationship would have an identifier 'R3345-1', the second 'R3345-2', and so on. The exact method of concatenation is up to your implementation (here a '-' character is used), just make sure it will not result in an identifier that could be confused with a Record entity.
- The value of the anzs:5478:NameWords element is the type of relationship. Standard values that can be used are: 'Contains'; 'Controls'; and 'Succeeds'. Other values can be used if required. Note that the standard values have a direction (e.g. One Record entity 'Contains' another Record Entity). Whether this Record entity is the 'from' or 'to' record entity is given by the anzs5478:RelationshipRole element.
- Relationships have a date/time when they were established. This date/time is contained in the value of the anzs5478:StartDate element. If the date/time the relationship was created is not known, use the date/time the younger of the two related Record entities was created.
- Relationships may also have an end date/time – this represents the date/time that a relationship between two Record entities was terminated. This is not shown in the sample XML snippet but takes the same form as the anzs5478:StartDate element, but uses the anzs:EndDate element. If your system records information about terminated relationships these relationships should be included, with the anzs5478:EndDate. However, we do not expect heroic efforts to reconstruct this information (e.g. by processing audit logs).
- The value of the anzs5478:AssignedEntityID element identifies the related Record entity. It should be identical to the value used in the anzs5478:IdentifierString element of the related Record entity.

- The value of the anzs5478:RelationshipRole is either '1' or '2', and this indicates the direction of the relationship. Relationships are often asymmetric (e.g. Record entity 1 'contains' Record entity 2). If the value of this element is '1' this indicates that the related Record Entity is the first (or from) entity in the relationship (e.g. the containing Record entity). If the value is '2' this indicates the second (or to) entity in the relationship.

4.7.2 Relationships with AS/NZS5478 Agent, Business, or Mandate entities

A Record entity normally has relationships with Agent entities (i.e. people or organisations that are associated with the Record entity (e.g. owning the Record entity, creating the Record entity). The Record entity may also have relationships with Business and Mandate entities. The Record/Agent, Record/Business or Record/Mandate relationships are also represented using ASNZS 5478 Relationship entities. In this case, however, the linked entity is included in the anzs5478:RelatedEntity element. The resulting XML has the form:

```
<anzs5478:Record rdf:parseType="Resource">
  [Other ANZS5478 metadata]
  <anzs5478:Relationship rdf:parseType="Resource">
    <anzs5478:EntityType rdf:datatype="xs:string">Relationship</anzs5478:EntityType>
    <anzs5478:Category rdf:datatype="xs:string">Provenance relationship</anzs5478:Category>
    <anzs5478:Identifier rdf:parseType="Resource">
      <anzs5478:IdentifierString rdf:datatype="xs:string">
        [Identifier of relationship]
      </anzs5478:IdentifierString>
    </anzs5478:Identifier>
    <anzs5478:Name rdf:parseType="Resource">
      <anzs5478:NameWords rdf:datatype="xs:string">
        [Type of Relationship]
      </anzs5478:NameWords>
    </anzs5478:Name>
    <anzs5478:DateRange rdf:parseType="Resource">
      <anzs5478:StartDate rdf:datatype="xs:datetime">
        [Date/time relationship was established]
      </anzs5478:StartDate>
    </anzs5478:DateRange>
    <anzs5478:RelatedEntity rdf:parseType="Resource">
  <anzs5478:Agent rdf:parseType="Resource">
    <anzs5478:EntityType rdf:datatype="xs:string">Agent</anzs5478:EntityType>
    <anzs5478:Category rdf:datatype="xs:string">Item</anzs5478:Category>
    <anzs5478:Identifier rdf:parseType="Resource">
      <anzs5478:IdentifierString rdf:datatype="xs:string">
        [Your system identifier for the Agent]
      </anzs5478:IdentifierString>
    </anzs5478:Identifier>
    <anzs5478:Name rdf:parseType="Resource">
      <anzs5478:NameWords rdf:datatype="xs:string">
        [The title of the agent entity (e.g. person name)]
      </anzs5478:NameWords>
    </anzs5478:Name>
    <anzs5478:DateRange rdf:parseType="Resource">
      <anzs5478:StartDate rdf:datatype="xs:datetime">
        [A date/time associated with the agent (e.g. when the agent was registered in system)]
      </anzs5478:StartDate>
  </anzs5478:RelatedEntity>
</anzs5478:Agent>
</anzs5478:Relationship>
</anzs5478:Record>
```

```

</anzs5478:DateRange>
</anzs5478:Agent>
  <anzs5478:RelationshipRole rdf:datatype="xs:string">
    1
  </anzs5478:RelationshipRole>
</anzs5478:RelatedEntity>
</anzs5478:Relationship>
</anzs5478:Record>

```

Notes:

- Again, only the metadata related to the relationship is included in this XML snippet.
- Comments on the value of the anzs5478:Relationship/anzs5478:Identifier, anzs5478:Relationship/anzs5478:Name & anzs5478:Relationship/anzs5478:DateRange elements are the same as in section 4.7.1.
- The value of anzs5478:Relationship/anzs:5478:Category can be 'Provenance Relationship' (as shown in the example) or 'Recordkeeping Event'.
- The value of the anzs5478:Relationship/anzs5478:Name/anzs:5478:NameWords element is the type of relationship. Standard values that can be used when anzs5478:Relationship/anzs:5478:Category is 'Provenance Relationship' are: 'Controls'; and 'Owns'. Standard values that can be when anzs5478:Relationship/anzs:5478:Category is 'Recordkeeping Event' are given in table D4.2 of the National Archives Australian Government Recordkeeping Metadata Standard. Other values can be used if required. Note that the standard values have a direction (e.g. One Record entity 'Contains' another Record Entity). Whether this Record entity is the 'from' or 'to' record entity is given by the anzs5478:RelationshipRole element.
- The value of the anzs5478:RelatedEntity element contains the RDF/XML representation of the related entity (an Agent in this case).
- The value of the anzs5478:RelationshipRole is either '1' or '2', and this indicates the direction of the relationship. Relationships are often asymmetric (e.g. Agent 1 'controls' Record entity 2). If the value of this element is '1' this indicates that the related Record Entity is the first (or from) entity in the relationship (e.g. the containing Record entity). If the value is '2' this indicates the second (or to) entity in the relationship.

5 General XML Metadata

This section states how a metadata package containing custom XML metadata is to be represented in a VEO.

The intended use of this mechanism is to allow custom metadata to be included as XML in a metadata package in a VEO without prior definition or arrangement. This allows inclusion of metadata from other metadata standards, and unstandardised system metadata from record systems.

Most record products contain collections of metadata about the records they hold; such as audit or history logs. The mechanism described in this section allows this metadata to be included in a VEO with minimal effort by a vendor.

We particularly encourage the inclusion of custom metadata configured in products at the request of customers.

5.1 Standards

The metadata must be expressed as valid [Extensible Markup Language \(XML\) 1.0 \(Fifth Edition\)](#). The XML representing the metadata is not validated against any schema.

5.2 XML Metadata Package preamble

Every Metadata Package containing AGLS metadata must commence and end as follows:

```
<vers:MetadataPackage>
  <vers:MetadataSchemaIdentifier>
    [URL identifying schema]
  </vers:MetadataSchemaIdentifier>
  <vers:MetadataSyntaxIdentifier>
    http://www.w3.org/TR/2008/REC-xml-20081126/8
  </vers:MetadataSyntaxIdentifier>
  [Record metadata...]
</vers:MetadataPackage>
```

Notes:

- The indentation is optional but makes the XML easier to read. If indentation is used, spaces are preferred as they do not take up as much horizontal space.
- The value of the `vers:MetadataSchemaIdentifier` element is a URL. The recommended values that should be used in this element are discussed in the section 5.3.
- The value of the `vers:MetadataSyntaxIdentifier` is `'https://www.w3.org/TR/2008/REC-xml-20081126/'`. Note the final slash.

⁸ This URL does need not link to a web page

5.3 The value of the Metadata Schema Identifier

The value of the `vers:MetadataSchemaIdentifier` element identifies the type of collection of metadata in this package. This could be the identity of the standard the collection conforms to, or (if it does not conform to a standard) the record system product that collected and generated the metadata package.

The following notes provide guidance as to selection of the value:

- The value is always a URL (Uniform Resource Locator).
- If the collection of metadata in this metadata package conforms to an external standard (e.g. Image metadata), the URL should be that of the formal specification (including version) of the standard used.
- If the collection of metadata does not conform to an external standard (i.e. it is internal metadata from the record system product), the URL should identify the vendor product (including version) that generated the metadata. The URL should not include any text that implies it forms part of a PROV namespace (e.g. the string 'PROV' or 'prov').
- The URL is only used as an identifier; no attempt is used to open the URL. It consequently does not need to point to a web page.
- All instances of this metadata package, in all information objects in all VEOs, should use the same URL to identify the same metadata collections.
- A `vers:MetadataSchemaIdentifier` can only appear once and can only contain one URL. If the metadata is drawn from several different sources, the metadata should be separated into multiple metadata packages.

5.4 Example of XML metadata

The package of XML metadata is almost freeform. It must be valid XML, but we do not check conformance against an XML schema.

If the package is based on a standard, the XML representation specified in the standard should be used.

If the package is custom set of metadata based on the information available in a product, we recommend creating a list of XML elements where the element name describes the metadata property, and the element value is the metadata property.

An example collection of XML metadata that could be generated from a vendor product is:

```
<DateRegistered>20/12/2013 at 10:31 AM</DateRegistered>
<DestructionDate>Destroy : No Triggers active</DestructionDate>
<UserDefinedField name="HasDeleteCaveat?">No</UserDefinedField>
<UserDefinedField name="Reason for Deletion"></UserDefinedField>
<UserDefinedField name="Paper File Exists?">Yes</UserDefinedField>
```

Notes:

- XML namespaces may occur, but the namespace 'vers:' cannot be used.
- Elements may be structured as required.
- Element names should be chosen to be meaningful to human readers

6 Standard aspects of RDF/XML

This section describes standard features of representing information in RDF or XML

6.1 Metadata expressed as RDF

All metadata expressed as RDF must have the `MetadataSyntaxIdentifier` property set to: `'http://www.w3.org/1999/02/22-rdf-syntax-ns#'` (without the quotes).

6.2 The value of the `rdf:about` attribute

RDF is intended to describe real world objects (known as Subjects). In RDF the real world object (subject) is identified using the `'rdf:about'` attribute. The value of an `'rdf:about'` attribute is a URI (Uniform Resource Identifier), typically a URL (Uniform Resource Locator – a web page address).

When generating the RDF metadata to include in Metadata Packages, it is necessary to generate URIs to identify the entities being described. These URIs must be:

- Globally unique. When imported into PROV's system, they must not conflict with any other URI held in our system.
- Persistent. An entity in an installed instance of your product should generate the same URI every time a VEO is generated containing that entity.

If the objects (subjects) in your system already have globally unique and persistent identifiers (e.g. a DOI), then you can use that identifier as the value of the `rdf:about` attribute.

If your identifier is only unique within the installed instance of your product, we recommend you use a URI of the form: `<staticPrefix><seriesConsignmentId>/"<uniqueId>` where:

- `<staticPrefix>` is an arbitrary string that is normally set in the product configuration on PROV's advice. The default value is `"https://prov.vic.gov.au/archive/VPRS"`, however it should be possible for the creator to change this string before generating the VEOs.
- `<seriesConsignmentId>` is an arbitrary string that is normally entered by the creator generating the VEOs. This string is obtained from PROV as part of the transfer negotiations. Normally this would simply be a positive integer (e.g. '421'), but we could be varied (e.g. to include consignments '421/P0'). The default value should be 'TRIAL' - this makes a valid URI but is clear that it is not a real transfer.
- `<uniqueId>` is the locally unique identifier in your system for the entity represented by this Information Object. The value is not specified here (subject to the rules of URIs), but the identifier should also be replicated as an `anzs5478:IdentifierString` property.

An example would be `"https://prov.vic.gov.au/archive/VPRS11223/1.401164"` where `"https://prov.vic.gov.au/archive/VPRS"` comes from the configuration, `"11223"` is entered by the creator before creating the VEOs, and `"1.401164"` is the locally unique identifier for the entity from the system.

6.3 ISO 8601 date/time values

Times and dates in VERS are encoded using in the following form (a subset of ISO 8601):

YYYY-MM-DDThh:mm:ssp[tt:mm]

Where:

- YYYY is a four digit year (e.g. '2024')
- MM is a two digit month (e.g. '03') ranging from '01' to '12'. Note months before October must have a leading zero.
- DD is a two digit day (e.g. '06') ranging from '01' to '31'. Note days below '10' must have a leading zero.
- hh is a two digit hour (e.g. '04', or '16') ranging from '00' to '24'. Note that 1) hours below '10' must have a leading zero; 2) twenty four hour clock is used; and 3) midnight can be either '00' or '24'. Times after midnight must have an hour of '00' (i.e. one second after midnight is represented as '00:00:01')
- mm is a two digit minute (e.g. '05') ranging from '00' to '59'. Note that minutes below '10' must have a leading zero.
- ss is a two digit second (e.g. '06') ranging from '00' to '60'. Note that seconds below '10' must have a leading zero. A '60' will only appear as a leap second. Note that fractions of a second are not permitted.
- p is either a '+', '-', or 'Z'. This marks the start of the timezone. 'Z' indicates that the time is represented in GMT.
- tt is a two digit timezone hour offset ranging from '00' to '14'. Note that seconds below '10' must have a leading zero.
- uu is a two digit timezone minute offset ranging from '00' to '59'. Note that seconds below '10' must have a leading zero.
- The characters '-', 'T', & ':' appear literally in the date/time.
- The characters '[' and ']' do NOT appear literally in the date/time. They indicate that the numeric timezone does not appear if the timezone indicator is 'Z'.

As an example, the current date/time when this document was written is expressed as "2024-03-06T15:10:04+11:00".

Most programming languages have routines to directly convert from the internal date/time representation to ISO 8601 format date/times.

6.3.1 Restricted accuracy date/times

It is acceptable to restrict the precision of a date/time:

- Where the time of day is not known, the time and timezone can be omitted (e.g. "2024-03-06")
- Where only the month and year is known, the date can be omitted (e.g. "2024-03")
- Where only the year is known, the month and date can be omitted (e.g. "2024")

6.3.2 RDF and restricted accuracy date/times

When expressed in XML, RDF values are tagged with a data type. The data type of a date/time is normally 'xs:dateTime'.

Where a restricted accuracy date/time is used, the following data types should be used instead of 'xs:dateTime':

- Where only a date is given (e.g. "2024-03-06") use 'xs:date'
- Where only the month and year is given (e.g. "2024-03") use 'xs:gYearMonth'
- Where only the year is given (e.g. "2024") use 'xs:gYear'

6.4 XML value and attribute encoding

Note: This section is only of relevance if you are generating XML from primitive text generation functions (e.g. C's 'printf', or Java's 'print()'). If you are generating XML using an XML aware library, the encodings discussed in this section will be performed automatically for you. Do not encode the characters manually if using an XML library.

XML cdata (character data) and attribute values require a small number of characters to be encoded. (These characters have special meanings when parsing XML). The character encodings are as follows:

- '<' encode as '<'
- '>' encode as '>'
- '&' encode as '&'
- '"' encode as '"'
- "'" encode as '''

For example, if the metadata value is "'Left & Right'" (note that the value contains single quote characters), the actual text in the XML file will be "'Left & Right'".

If you are generating XML using primitive text printing commands, you must explicitly check for the presence of these five characters and replace them with the encoded values. If you are generating XML using an XML library this encoding will be performed by the library.

In addition, XML cannot contain any non Unicode characters. This specifically affects the ASCII control characters (i.e. characters 0 to 31), except #x9 (tab), #xA (line feed), and #xD (carriage return). In the unlikely event that your value contains ASCII control characters (except tab, line feed, or carriage return), you must remove them from the value.

Appendix 1 AGLS Metadata (Deprecated)

Important Note: The use of AGLS metadata in Metadata Packages is deprecated. New implementations of this specification must use AS/NZS 5478 metadata to represent the standard metadata. PROV will continue to accept VEOs containing AGLS metadata from existing implementations. This section is retained for historical purposes.

This section states how a metadata package containing AGLS metadata is to be represented in a VEO.

Appendix 1.1 Standards

Victorian public offices must use an augmented version of the Victorian government profile of AGLS. The Victorian AGLS profile is described in the [AGLS Victoria Metadata Implementation Manual](#)⁹. The current version of this document is Version 4.0 issued July 2011.

The AGLS metadata must be expressed as RDF. The method of representing AGLS using RDF described in [AGLS Metadata Standard, Guide to expressing AGLS metadata in RDF](#)¹⁰ must be used. The final version is Version 1.1 issued January 2022.

Appendix 1.2 Additional AGLS Properties

The following additional AGLS properties **must** be added when creating a VEO that is to be archived within a public office¹¹. These properties document the planned disposal of the VEO.

It is not necessary to add these properties when creating a VEO that is to be immediately transferred to PROV¹². This is because the VEO is being created as part of the disposal process.

Disposal Review Date		
OBLIGATION: Conditional		
Sub-properties	Notes on use	Rendered
N/A	One of DISPOSAL REVIEW DATE or DISPOSAL CONDITION must be present for resources held in agencies.	versterms: disposalReviewDate

⁹ This is available at <https://www.vic.gov.au/sites/default/files/2019-06/AGLS-Victoria-July-2011-Version-4.PDF>

¹⁰ This is available at https://web.archive.org/awa/20230802044920mp_/https://agls.gov.au/pdf/Guide%20to%20expressing%20AGLS%20metadat a%20in%20RDF%20v1.1.PDF

¹¹ The DISPOSAL properties are required on records held in the public office so that records managers can be cued at the appropriate time to determine if the information contained in the VEO can be disposed of.

¹² Transferring records to PROV is a disposal action. Hence, when creating VEOs specifically for immediate transfer to PROV the disposal decision has already been taken. These disposal fields are consequently redundant.

	<p>Use DISPOSAL REVIEW DATE when the disposal decision point can be reduced to a simple date that is known in advance (e.g. if the disposal condition is 'Dispose 7 years after creation', and the resource was created in 2020, the DISPOSAL REVIEW DATE will be 2027).</p> <p>The result of the review may be to retain the resource for a further period, in which case the DISPOSAL REVIEW DATE must be updated to indicate when the decision is to be revisited.</p> <p>If the resource is permanent, this property indicates when a decision should be made to transfer it to PROV.</p> <p>Dates must be represented using ISO8601 but should only specify the year unless very precise disposal is required.</p>	
--	---	--

Disposal Condition		
OBLIGATION: Conditional		
Sub-properties	Notes on use	Rendered
N/A	<p>One of DISPOSAL REVIEW DATE or DISPOSAL CONDITION must be present for resources held in public offices.</p> <p>Use DISPOSAL CONDITION when the disposal decision point cannot be reduced to a simple date (e.g. 'dispose of 30 years after use ceases').</p> <p>DISPOSAL CONDITION contains a textual description of the conditions under which this resource can be disposed of (this would normally be a disposal trigger and retention period).</p>	versterms: disposalCondition

Disposal Action		
OBLIGATION: Conditional		
Sub-properties	Notes on use	Rendered
N/A	<p>DISPOSAL ACTION must be present for resources held in public offices.</p> <p>If the resource is permanent, this property must contain the value 'Retain as State Archives' 'Transfer to PROV'.</p> <p>If the resource is temporary, this property contains a suggestion as to the fate of the information. Methods of disposal could include sale, transfer to another organisation, or destruction. This suggestion, however, need</p>	versterms: disposalAction

	not be followed when the agency makes the decision to dispose of the resource.	
--	--	--

Disposal Reference		
OBLIGATION: Conditional		
Sub-properties	Notes on use	Rendered
N/A	DISPOSAL REFERENCE must be present for resources held in public offices. Use to contain a reference to the Retention and Disposal Authority clause that states the retention period governing this class of resource.	versterms: disposalReference

Appendix 1.3 AGLS Metadata Package preamble

Every Metadata Package containing AGLS metadata must commence and end as follows:

```
<vers:MetadataPackage xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <vers:MetadataSchemaIdentifier>
    http://www.vic.gov.au/blog/wp-content/uploads/2013/11/AGLS-Victoria-2011-V4-Final-2011.pdf
  </vers:MetadataSchemaIdentifier>
  <vers:MetadataSyntaxIdentifier>
    http://www.w3.org/1999/02/22-rdf-syntax-ns#
  </vers:MetadataSyntaxIdentifier>
  <rdf:RDF xmlns:dcterms="http://purl.org/dc/terms/"
    xmlns:agls="http://www.agls.gov.au/agls/terms/"
    xmlns:versterms="http://www.prov.vic.gov.au/vers/terms/">
    <rdf:Description rdf:about="[Record Entity Identifier]">
      [Record metadata...]
    </rdf:Description>
  </rdf:RDF>
</vers:MetadataPackage>
```

Notes:

- The indentation is optional but makes the XML easier to read. If indentation is used, spaces are preferred as they do not take up as much horizontal space.
- The xmlns:rdf namespace is defined as an attribute in the vers:MetadataPackage element
- The value of the vers:MetadataSchemaIdentifier element is '<http://www.vic.gov.au/blog/wp-content/uploads/2013/11/AGLS-Victoria-2011-V4-Final-2011.pdf>'. This URL is an identifier and does not point to a web page.
- The value of the vers:MetadataSyntaxIdentifier is '<http://www.w3.org/1999/02/22-rdf-syntax-ns#>'.

- The xmlns:dcterms, xmlns:aglstterms, & xmlns:verstems namespaces are defined as attributes in the rdf:RDF element.
- The rdf:about attribute in the rdf:Description element contains a unique identifier for the record. The value for this identifier is described in section 6.2.

Appendix 1.4 Basic record metadata

Each AGLS metadata package contains exactly one collection of related metadata. Mapped to XML, a record with the bare minimum of metadata has the following form. Text in **bold** is replaced by values from your system.

```
<rdf:Description rdf:about="[Record Entity Identifier]">
  <dcterms:creator>Creator1</dcterms:creator>
  <dcterms:date rdf:datatype="xsd:date">
    [The date/time the record entity was created in ISO 8601 form]
  </dcterms:date>
  <dcterms:title>
    [The title of the record entity]
  </dcterms:title>
  <dcterms:identifier>
    [Your system identifier for the Record entity]
  </dcterms:identifier>
  <dcterms:publisher>
    [The organisation responsible for managing this record entity]
  </dcterms:publisher>
  <dcterms:description>
    [A description of this record entity]
  </dcterms:description>
</rdf:Description>
```

Notes:

- It is expected that the AGLS description generated from a real record system will contain much more metadata than is shown here.
- The metadata elements are immediately underneath the rdf:Description element.
- All XML elements that DO NOT have subelements (i.e. all of the AGLS metadata elements) have an rdf:datatype element that contains the data type of the element value. This is normally 'xs:string', but in some cases will be different. The actual data type to be used for each AN/NZS 5478 property is shown in the third column in the table in section Appendix 1.5.
- The value of the dcterms:identifier element will often be included in the rdf:about attribute (see section 6.2)
- The value of the dcterms:date is an ISO 8601 date/time. The exact format of an ISO 8601 date/time is described in section 6.3.

Appendix 1.5 Mapping AGLS Property names to XML elements

The following mapping between AGLS properties and RDF property names must be used. The AGLS documents should be consulted for the meanings of the properties, and whether the properties must or may be present, and whether they can be repeated.

AGLS Property		XML Element Name	RDF Data Type
4.1.1	Creator	dcterms:creator	xs:string
4.1.2	Date	dcterms:date	xs:date
4.1.3	Title	dcterms:title	xs:string
4.2.1	Availability	aglstterms:availability	xs:string
4.2.2	Identifier	dcterms:identifier	xs:string
4.2.3	Publisher	dcterms:publisher	xs:string
4.3.1	Description	dcterms:description	xs:string
4.3.2	Function	aglstterms:function	xs:string
4.3.3	Language	dcterms:language	xs:string
4.3.4	Subject	dcterms:subject	xs:string
4.3.5	Type	dcterms:type	xs:string
4.4.1	Access Rights	dcterms:accessRights	xs:string
4.4.2	Act	aglstterms:act	xs:string
4.4.3	Audience	dcterms:audience	xs:string
4.4.4	Aggregation Level	aglstterms:aggregationLevel	xs:string
4.4.5	Alternative	dcterms:alternative	xs:string
4.4.6	Available	dcterms:available	xs:string
4.4.7	Bibliographic Citation	dcterms:bibliographicCitation	xs:string
4.4.8	Case	aglstterms:case	xs:string
4.4.9	Category	aglstterms:category	xs:string
4.4.10	Conforms To	dcterms:conformsTo	xs:string
4.4.11	Contributor	dcterms:contributor	xs:string
4.4.12	Coverage	dcterms:coverage	xs:string
4.4.13	Available	dcterms:available	xs:string
4.4.14	Copyright	dcterms:dateCopyright	xs:string
4.4.15	Date Licensed	aglstterms:dateLicensed	xs:string
4.4.16	Document Type	aglstterms:documentType	xs:string
4.4.17	Extent	dcterms:extent	xs:string
4.4.18	Format	dcterms:format	xs:string
4.4.19	Has Format	dcterms:hasFormat	xs:string
4.4.20	Has Part	dcterms:hasPart	xs:string
4.4.21	Has Version	dcterms:hasVersion	xs:string
4.4.22	Is Based On	aglstterms:isBasedOn	xs:string
4.4.23	Is Basis For	aglstterms:isBasisFor	xs:string
4.4.24	Is Format Of	aglstterms:isFormatOf	xs:string
4.4.25	Is Part Of	dcterms:isPartOf	xs:string
4.4.26	Is Referenced By	dcterms:isReferencedBy	xs:string
4.4.27	Is Replaced By	dcterms:isReplacedBy	xs:string
4.4.28	Is Required By	dcterms:isRequiredBy	xs:string

AGLS Property		XML Element Name	RDF Data Type
4.4.29	Issued	dcterms:issued	xs:string
4.4.30	Is Version Of	dcterms:isVersionOf	xs:string
4.4.31	Jurisdiction	aglsterms:jurisdiction	xs:string
4.4.32	License	dcterms:license	xs:string
4.4.33	Mandate	aglsterms:mandate	xs:string
4.4.34	Medium	dcterms:medium	xs:string
4.4.35	Modified	dcterms:modified	xs:string
4.4.36	Protective Marking	aglsterms:protectiveMarking	xs:string
4.4.37	References	dcterms:references	xs:string
4.4.38	Regulation	aglsterms:regulation	xs:string
4.4.39	Relation	dcterms:relation	xs:string
4.4.40	Replaces	dcterms:replaces	xs:string
4.4.41	Requires	dcterms:requires	xs:string
4.4.42	Rights	dcterms:rights	xs:string
4.4.43	Rights Holder	dcterms:rightsHolder	xs:string
4.4.44	Service Type	aglsterms:serviceType	xs:string
4.4.45	Source	dcterms:source	xs:string
4.4.46	Spatial	dcterms:spatial	xs:string
4.4.47	Temporal	dcterms:temporal	xs:string
4.4.48	Valid	dcterms:valid	xs:string

Appendix 1.6 AGLS controlled values

A number of the properties in AGLS have controlled values which are referred to as Encoding Schemes. In representing these values in VERS, the text used in the XML value must be identical to that presented in AGLS specifications with no leading or trailing spaces.

Appendix 1.7 Expressing AGLS property values in XML

AGLS property values expressed in XML must be XML safe. The necessary encoding is described in section 6.4.

Appendix 1.8 Expressing AGLS relationships

AGLS relationships are expressed using the Relation (dcterms:relation) element, or one of its sub-properties (Conforms To, Has Format, Has Part, Has Version, Is Based On, Is Basis For, Is Format Of, Is Part Of, Is Referenced By, Is Replaced By, Is Required By, Is Version Of, References, Replaces, or Requires). The value of any of these properties is the value of the Identifier property in the related Information Object.

Copyright Statement

© State of Victoria 2025



Except for any logos, emblems, and trademarks, this work is licensed under a Creative Commons Attribution 4.0 International license, to the extent that it is protected by copyright. Authorship of this work must be attributed to the Public Record Office Victoria. To view a copy of this license, visit <https://creativecommons.org/licenses/by/4.0/legalcode>

Disclaimer

The State of Victoria gives no warranty that the information in this version is correct or complete, error free or contains no omissions. The State of Victoria shall not be liable for any loss howsoever caused whether due to negligence or otherwise arising from the use of this Specification.