

Maritime Limits and Boundaries Product Specification

Annex B: Data Product Format (Encoding)

Edition 1.0.0 – October 2019

IHO



International
Hydrographic
Organization

Published by the
International Hydrographic Organization
4b quai Antoine 1^{er}
Principauté de Monaco
Tel: (377) 93.10.81.00
Fax: (377) 93.10.81.40
info@ihoh.int
www.ihoh.int

© Copyright International Hydrographic Organization 2019

This work is copyright. Apart from any use permitted in accordance with the Berne Convention for the Protection of Literary and Artistic Works (1886), and except in the circumstances described below, no part may be translated, reproduced by any process, adapted, communicated or commercially exploited without prior written permission from the International Hydrographic Organization (IHO). Copyright in some of the material in this publication may be owned by another party and permission for the translation and/or reproduction of that material must be obtained from the owner.

This document or partial material from this document may be translated, reproduced or distributed for general information, on no more than a cost recovery basis. Copies may not be sold or distributed for profit or gain without prior written agreement of the IHO Secretariat and any other copyright holders.

In the event that this document or partial material from this document is reproduced, translated or distributed under the terms described above, the following statements are to be included:

"Material from IHO publication [reference to extract: Title, Edition] is reproduced with the permission of the IHO Secretariat (Permission No/...) acting for the International Hydrographic Organization (IHO), which does not accept responsibility for the correctness of the material as reproduced: in case of doubt, the IHO's authentic text shall prevail. The incorporation of material sourced from IHO shall not be construed as constituting an endorsement by IHO of this product."

"This [document/publication] is a translation of IHO [document/publication] [name]. The IHO has not checked this translation and therefore takes no responsibility for its accuracy. In case of doubt the source version of [name] in [language] should be consulted."

The IHO Logo or other identifiers shall not be used in any derived product without prior written permission from the IHO Secretariat.

Document Control

Changes to this document are coordinated by the IHO S-121 Project Team (S-121PT) which is a subsidiary group of the S-100 Working Group (S-100WG). New editions will be made available via the IHO web site.

Edition Number	Date	Author	Purpose
Edition 1.0.0	April 2018	S-121PT	Approved by HSSC 10 under the new review cycle for testing. (Reference; HSSC10/16)
Edition 1.0.1	January 2019	S-121PT	Major revision based on testing to and feedback from UN DOALOS to address text encoding for deposit with the UN. Reviewed and accepted as suited to purpose by UN DOALOS December 2018.
Ballot version of Edition 1.0.0	February 2019	IHO Sec	Revised format and structure.
Edition 1.0.0	September 2019	S-121PT	Revised as per results of ballot comment review meeting August 2019

Page intentionally left blank

TABLE OF CONTENTS

	Overview	1
1	Introduction	1
1.1	System Dependant Exchange File	2
1.2	Explicit Textual Format for list of Geographic Coordinates for Deposit.....	2
1.2.1	Support for Metadata	3
2	Implementation Schema.....	5
3	Explicit Text Encoding Format.....	9
3.1	Explicit Text Encoding Structure.....	9
3.2	Cross referencing	10
3.3	Delimiting Structure	11
3.3.1	Record Structure.....	12
3.3.2	Data Blocks.....	13
3.3.3	Data File	15
3.3.4	Block Descriptor Record	17
3.3.5	Data Block Groups.....	19
3.3.6	Table Structure	19
3.4	Data Types	20
3.5	Information Elements.....	21
3.5.1	Metadata Block	22
3.5.2	Governance Block	24
3.5.3	Basic Administrative Unit Block	25
3.5.4	Rights Restrictions and Responsibilities Blocks.....	25
3.5.5	Party Group Blocks.....	26
3.5.6	Source Block.....	27
Appendix A	Implementation Schema	35
Appendix B	The Carriage Return / Line Feed Delimiter	49
Appendix C.	Abstract Syntax Notation	51
C 1.	Introduction	51
C 2.	Syntactic Structure.....	51
C 2.1.	Assignment	52
C 2.2.	SEQUENCE and CHOICE	52
C 2.3.	OPTIONAL and DEFAULT	52
C 2.4.	MACRO	52
C 2.5.	Tag Numbers	52
C 2.6.	Grammar.....	53
Appendix D.	Profile for Deposit with the UN	55
D 1.	Profile File Structure	55
D 2.	Profile Metadata Block.....	55
D 3.	Profile Governance Block	56
D 4.	Profile Party Block Group	57
D 5.	Profile Basic Administrative Unit Block.....	58
D 6.	Profile Right Restriction and Responsibility Blocks	58
D 6.1.	Profile Rights Block.....	58
D 6.2.	Profile Restrictions Block	59
D 6.3.	Responsibility Block.....	60
D 7.	Profile Source Block	61
D 8.	Profile Zone Block and Surface Block Group	61
D 8.1.	Profile Zone Block.....	62
D 8.2.	Profile Surface Block	63
D 9.	Space Blocks	63
D 10.	Profile Limit Block and Curve Block Group	63
D 10.1.	Profile Limit Block	64
D 10.2.	Profile Curve Block	65
D 11.	Profile Location and Point Blocks	66

D 11.1.	Profile Location Block	66
D 11.2.	Profile Point Block.....	67
D 12.	Combined Location Object and Point Object Block.....	68
Appendix E.	Record Oriented Encoding Structure in Abstract Syntax Notation ASN.1	71
Appendix F.	Bibliography	77

Overview

The S-121 Maritime Limits and Boundaries Product Specification defines the structure and relationships of data elements that are used to describe Maritime Limits and Boundaries. This document describes the interfaces through which external processes and users can access MLB data. For the deposit with the United Nations under the UN Convention on the Law of the Sea (UNCLOS) and the distribution in digital form of Maritime Limits and Boundaries information, a specialized human readable encoding is needed. This explicit text encoding is described in this document. This encoding format must not only be human readable, and simple enough to be read by all interested parties, it must also be parseable so that it can also be read back into a computer system. Whereas a Maritime Limits and Boundaries data base must be capable of supporting all of the structures and relationships defined in the S-121 Product Specification, it is not necessary that all of these elements and relationships be presented in any particular encoding output or input. The internal maintenance structure of the data does not need to be exposed. In the case of deposit with the UN only those elements required to be deposited are included in the output format.

1 Introduction

Encoding is the translation of information from a structure, such as an internal database, into a format that can be communicated. The converse is decoding in which the transmitted information is converted into a form that can be used in the end application. Encoding establishes the interface at which data can be used.

Encoding is extremely important for S-121 Marine Limits and Boundaries to support the deposit of the description of states sovereign boundaries to the UN DOALOS in compliance with UNCLOS. As such MLB data must be easily readable; this means that the data must be available in a human readable form. As stated in the United Nations Convention on the Law of the Sea (UNCLOS) this form could be "Charts" that can be easily viewed or it could be "Lists of Geographical Coordinates" that can be easily read. The output format required to support machine and human readable "Lists of Geographical Coordinates" must be explicit and easy to read by non-technical experts in a legislative, administrative or other similar environments; and able to be used to support deposit with the Division for Ocean Affairs and the Law of the Sea (DOALOS) of the Office of Legal Affairs in the UN. The output needs to look closely like the information that is printed in treaties or national laws. This output must be textual so that there are no barriers to its understanding. It also needs to be simple.

The Extensible Markup Language (XML) is not an appropriate human readable format for this application. It is not simple and even Information Technology experts get lost in all of the nested brackets, XLink¹ references and other complexities of the XML syntax.

A structured text encoding is described in the following sub clauses that meets the requirements of being explicit and easy to read by non-technical experts. There may be multiple different textual output formats that may be used by different states, which are variations on the "Explicit Text Format" defined here. With each such output format there needs to be a complementary input process to allow the data to be read into a computer system. The deposit of a data set using a structured text encoding with UN/DOALOS requires a standardized output that can be read by a common piece of reader software.

This document provides detail for an Explicit Text Format that is simple and easy to interpret in an official or legislative environment or to be used for deposit of a "List of Geographic Coordinates" to UN/DOALOS. The format allows Locations, Limits (and Boundaries) and Zones with their associated S-100 compliant geometric Points, Curves and Surfaces to be described.

Data may also be exchanged between states using existing published international standards from IHO, ISO and OGC. Only the general requirements for these encodings are given in this document.

¹ XML document reference linkage

No cartographic Portrayal is defined for the Explicit Text Format. Portrayal may be defined in other uses based on IHO S-100 portrayal methods.

1.1 System Dependant Exchange File

At times nation states or other organizations working with MLB data may wish to exchange whole data sets or major portions of databases between themselves. This can be done in the native format of the GIS system being used. For example, Esri Shape files can be used for communication between Esri systems and CARIS files can be used to exchange between CARIS systems. Inter system communication is dependent upon the capability of such systems to be able to read each other's files.

It is possible for a generic GML exchange schema to be defined addressing all of the S-121 schema and this could be used for inter system data exchange.

A generic GML schema for S-121 would not support shared attributes by reference as is used to describe the information objects in S-121. This is because generic GML does not support such structures. This means that in a full GML schema these attributes would need to be collapsed into ordinary attributes; that is, these attributes would need to be repeated as direct attributes of feature objects. A structure would need to be developed to address the features **BAUnit** and **Governance**. If the XML schema were extended to support such features using structures such as XLink, these information features could be handled, but generic GML presentation software would not understand such objects and would ignore them.

1.2 Explicit Textual Format for list of Geographic Coordinates for Deposit

MLB data that is output for deposit with the UN/DOALOS needs to be in an explicit and easy to read textual format. The output for deposit with the UN/DOALOS needs to contain all the information required by the UN Convention on the Law of the Sea for deposit. The output needs to look closely like the information that is printed in treaties or national laws. The content will be decided by the nation state.

Coastal States, under article 16, paragraph 2; article 47, paragraph 9; article 75, paragraph 2; and article 84, paragraph 2 of UNCLOS, are required to deposit with the Secretary-General of the United Nations charts showing straight baselines and archipelagic baselines as well as the outer limits of the territorial sea, the exclusive economic zone and the continental shelf, including the extended continental shelf beyond 200 metres depth. Alternatively, the lists of geographical coordinates of points, specifying the geodetic datum, may be substituted.

For the deposit of “a List of Geographic Coordinates” an explicit textual format is required where Latitude / Longitude coordinates are described explicitly as points “specifying the geodetic datum”. A geographic coordinate may be specified in different coordinate systems using different geodetic datums in different documents. Also, the same geographic coordinate may be specified in WGS84² or another world wide reference system. Only the geographic coordinate specified using the geodetic datum identified in the official legislative proclamation by a nation is considered to be correct. All other representations are derived from the official specification and may contain errors or conversion inaccuracies. Defining a common geodetic datum WGS84 or another world wide reference system is recommended.

At a minimum there needs to be some sort of header establishing title, date and other context; a description of which object is being described; and a set of Longitude / Latitude points describing the geometry supporting that object.

There exist three types of features – Locations, Limits, and Zones – defined in S-121. A fourth type of feature, a Space, may be defined by a state including an area and a height. For deposit with UN/DOALOS only the Locations, Limits, and Zones need to be addressed.

The central information feature in the S-121 schema is the “BasicAdministrativeUnit”. This feature may be related to one or more “FeatureUnit” features describing points, limits and or zones that are related. Associated with each BasicAdministrativeUnit is one or more “Governance” features. The governance feature describes the context. It includes the title, date, a description and other contextual attributes. Normally there is one Governance feature related to one BasicAdministrativeUnit, although more complex

² World Geodetic System 1984

structures are permitted in the S-121 data model such as a hierarchy of sets of BasicAdministrativeUnits and multiple Governance features. “FeatureUnit” features are simple geospatial features with both thematic and spatial attributes. The “RightsRestrictionsResponsibilities” and “Party” and “GroupParty” information features are effectively shared attributes by reference in the S-121 information schema. These can be flattened into simple attributes in an explicit textual exchange format or references may be used. References need to be limited because they are hard to follow when reading a printed output. The “Source” information feature is also a shared attribute by reference. Source references, when they are used, can be verbose and it is best to maintain these as attributes by reference in an exchange structure.

To support an explicit textual exchange format for deposit with UN/DOALOS it is necessary to define methods to support all of the elements in the S-121 schema; however, in practise only a few of these structures will be used. These structures are defined in later clauses in this document.

Generating explicit text is not difficult. It is essentially a number of formatted print statements (or XSLT³ transform clauses⁴) in the software generating the output. The difficult task is unambiguously reading the Explicit Text Formatted data back into a computer system. If a complex structure is described it will require pointers and delimiters to be able to identify the information elements. These pointers and delimiters eliminate the simplicity and are self-defeating. Their use must be minimized.

The approach adopted for deposit to UN/DOALOS is to require that deposits be only simple cases. This will mean that complex cases will need to be broken down into several simpler deposits. In general this is already what the common practise is for nation states. A generic piece of ingest software can then be developed that will only implement the simple cases.

There is a very wide diversity in how nation states describe their Maritime Limits and Boundaries in their own laws and proclamations.

The basic elements of the S-121 schema are the same and would be encoded in a text format in a similar manner.

1.2.1 Support for Metadata

S-100 requires Product Specifications to support a minimum set of metadata.

By default the character set encoding standard used should be UTF8 (the widely used internet common standard Unicode) which corresponds to ISO standard 10646-1. Even though this could be documented in the encoding schema for the particular output format it is best to always include it in the metadata. Similarly the language of the metadata and data is best documented in the metadata. Unless it is clear from the context, such as the fact that data is extracted from a particular nation states database, it is best to also include the contact information in the metadata.

Minimum elements as required by ISO are:

Name	Description	Path (in ISO 19115 standard)	Optionality (for data set exchange)	Data Type
Title	Name by which the cited resource is known.	MD_Metadata.identificationInfo > MD_DataIdentification.citation > CI_Citation.title	M	free text (CharacterString)
Abstract	Brief narrative summary of the dataset's contents	MD_Metadata.identificationInfo > MD_DataIdentification.abstract	M	free text (CharacterString)
Topic	The main theme(s) of the dataset from the ISO list of topic areas.	MD_Metadata.identificationInfo > MD_DataIdentification.topicCategory	M	CodeList : MD_TopicCategoryCode
Metadata contact name	Name of the individual responsible for the metadata.	MD_Metadata.contact > CI_ResponsibleParty.individualName	C (documented if 'organisationName' and 'positionName' not documented)	free text (CharacterString)
Metadata contact organisation	Organization responsible for the metadata.	MD_Metadata.contact > CI_ResponsibleParty.organisationName	C (documented if 'individualName' and 'positionName' not documented)	free text (CharacterString)

³ Extensible Stylesheet Language Transform

⁴ One strategy for generating Explicit Text output is to generate GML and then use an XSLT transform to produce text.

Metadata contact position	Position of the individual responsible for the metadata.	MD_Metadata.contact > CI_ResponsibleParty.positionName	C (documented if 'individualName' and 'organisationName' not documented)	free text (CharacterString)
Metadata contact role	Role of the individual responsible for the metadata.	MD_Metadata.contact > CI_ResponsibleParty.role > CI_RoleCode	C	Codelist CI_RoleCode
Reference date	reference date for the cited resource	MD_Metadata.identificationInfo > MD_DataIdentification.citation > CI_Citation.date > CI_Date.date	M	repeatable (Date and Codelist CI_DateTypeCode)
Dataset language	Languages of the dataset using standard ISO 639-2 three letter codes. Note: Three letter language code followed by an optional ISO 3166 three letter country code	MD_Metadata.identificationInfo > MD_DataIdentification.language	M	free text (CharacterString)
Metadata date	Metadata creation date	MD_Metadata.dateStamp	M	Date

(M) = mandatory (M*) = mandatory and repeatable

(C) = conditional (C*) = conditional and repeatable

(O) = optional (O*) = optional and repeatable

Additional metadata elements from the ISO Metadata Standard specified by S-100:

Name	Description	Path (in ISO 19115 standard)	Optionality (for data set exchange)	Data Type
Metadata file identifier	A unique phrase or string which uniquely identifies the metadata file	MD_Metadata.fileIdentifier	M	free text (CharacterString)
Metadata language	Language of the metadata composed of an ISO 639- 2/T three letter language code and an ISO 3166-1 three letter country code.	MD_Metadata.language	C (documented if not defined by the encoding process)	free text (CharacterString)
Metadata character set	character coding standard in the metadata.	MD_Metadata.characterSet	C (documented if ISO 10646-1, is not used and not defined by the encoding process)	CodeList MD_CharacterSetCode
Metadata file parent identifier	The unique name of the file or associated fileIdentifier, related in higher hierarchy to the file.	MD_Metadata.parentIdentifier	C (documented if the hierarchy of a higher level exists)	free text (CharacterString)
Metadata hierarchy level	Level to which the metadata applies. (e.g. dataset)	MD_Metadata.hierarchyLevel	O (assumed to be 'dataset' if MD_Metadata.hierarchyLevel is omitted)	CodeList MD_ScopeCode
Metadata hierarchy level name	Name of the level to which the metadata applies. (e.g. dataset)	MD_Metadata.hierarchyLevelName	O (assumed to be 'dataset' if MD_Metadata.hierarchyLevelName is omitted)	free text (CharacterString)
Dataset reference date type	Type of date	MD_Metadata.identificationInfo > MD_DataIdentification.citation > CI_Citation.date > CI_Date.dateType > CI_DateTypeCode	M	CodeList C_DateTypeCode
Dataset character set	Character coding standard in the dataset.	MD_Metadata.identificationInfo > MD_DataIdentification.characterSet	C (documented if ISO 10646-1 is not used)	CodeList MD_CharacterSetCode
Extent	Description of the geographic location of the dataset	MD_Metadata.identificationInfo > MD_DataIdentification.extent > EX_Extent > EX_GeographicDescription.geographicIdentifier > MD_Identifier.code	C (See notes 2 and 3)	free text (CharacterString)
West longitude	Describes the spatial coverage of the dataset.	MD_Metadata.identificationInfo > MD_DataIdentification.extent > EX_Extent > EX_GeographicBoundingBox.westBoundLongitude	C (See notes 2 and 3)	Decimal fields containing west coordinates of the bounding box
East longitude	Describes the spatial coverage of the dataset.	MD_Metadata.identificationInfo > MD_DataIdentification.extent > EX_Extent > EX_GeographicBoundingBox.eastBoundLongitude	C (See notes 2 and 3)	Decimal fields containing east coordinates of the bounding box

South latitude	Describes the spatial coverage of the dataset.	MD_Metadata.identificationInfo > MD_DataIdentification.extent > EX_Extent > EX_GeographicBoundingBox.southBoundLatitude	C (See notes 2 and 3)	Decimal fields containing south coordinates of the bounding box
North latitude	Describes the spatial coverage of the dataset.	MD_Metadata.identificationInfo > MD_DataIdentification.extent > EX_Extent > EX_GeographicBoundingBox.northBoundLatitude	C (See notes 2 and 3)	Decimal fields containing north coordinates of the bounding box

NOTE1 ISO 10646-1 - Information technology — Universal Multiple-Octet Coded Character Set

NOTE2 For a geographic dataset, include metadata for the geographic bounding box (West longitude, East longitude, South latitude and North latitude) or the geographic description identifier (The use of geographic bounding box is recommended).

NOTE3 If any one of west longitude, east longitude, south latitude or north latitude exists, then the remaining three must also be completed

2 Implementation Schema

The Implementation Model for the S-121 standard is described in Appendix B, clause B.2 of the S-121 Product Specification document. It is described in more detail here so that the various elements that need to be encoded can be identified.

The model in the Product Specification, Figure B-14 shows the major classes; but it also shows collection classes. The collection classes are abstract and make the model look simpler but are not directly implemented. Also, this model does not show the details of the S121_FeatureUnit and S121_SpatialAttributeType class.

The model in Figure 2-1 below shows all of the implementation classes without the collection classes except for the details of the S121_FeatureUnit and S121_SpatialAttributeType which are shown separately. Otherwise the figure would be too complex to read. Note that there are many more relationships shown when only the instantiable classes are exposed.

In an implementation, relationships are addressed as pointers so they become a type of attribute. The composition relationship between S121_Party and S121_GroupParty includes a relationship class. This is effectively an attribute on an attribute and can be implemented as two attributes, one indicating participation of a party in a group and the other optionally indicating a percentage share in the participation.

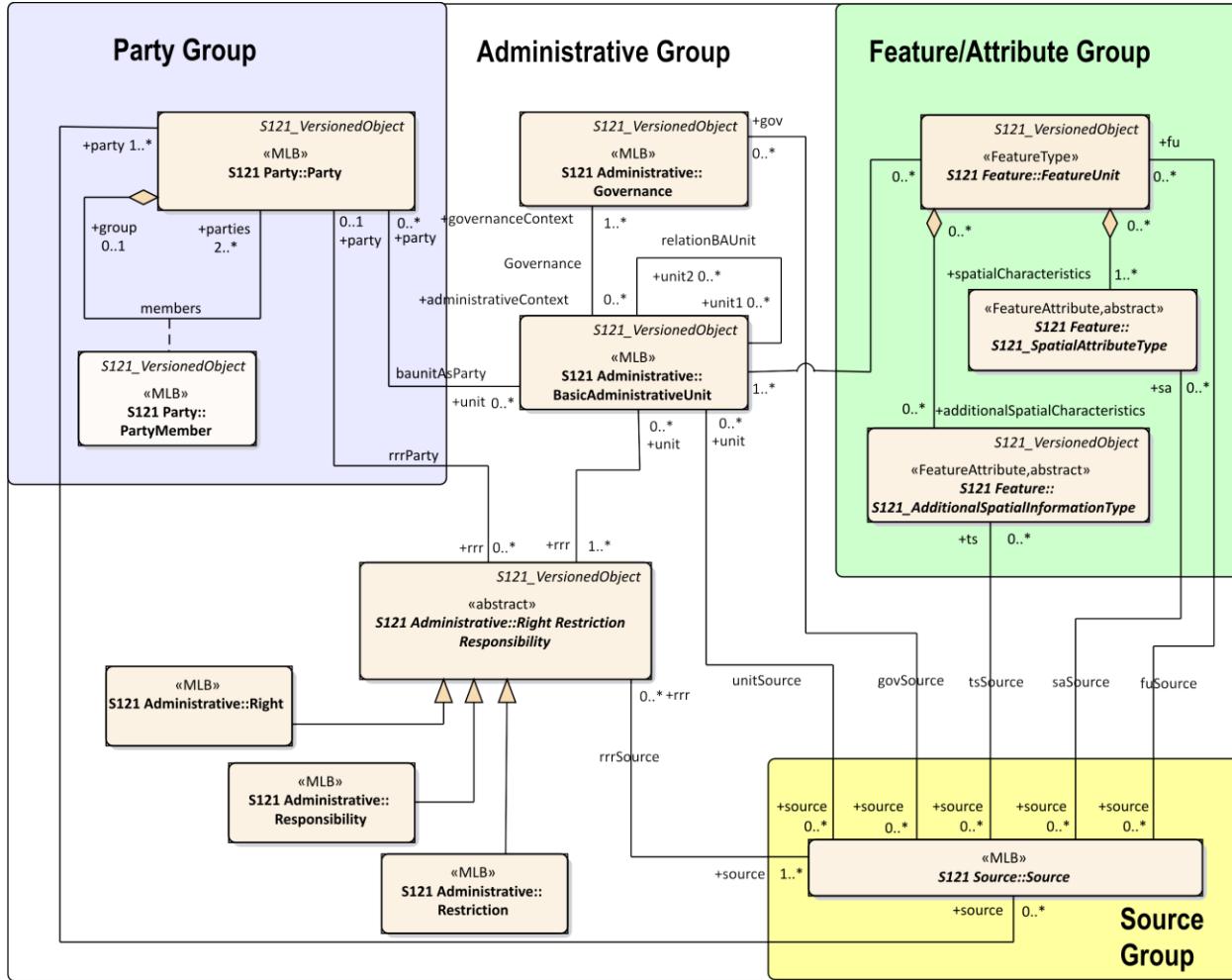


Figure 2-1 – S-121 Instantiated Implementation Model

The structure of the S121_FeatureUnit and S121_SpatialAttributeType is shown in Figure 2-2 below. Since each Feature Unit and each Spatial Attribute Type can have a source there are a lot of connectors to the Source Group.

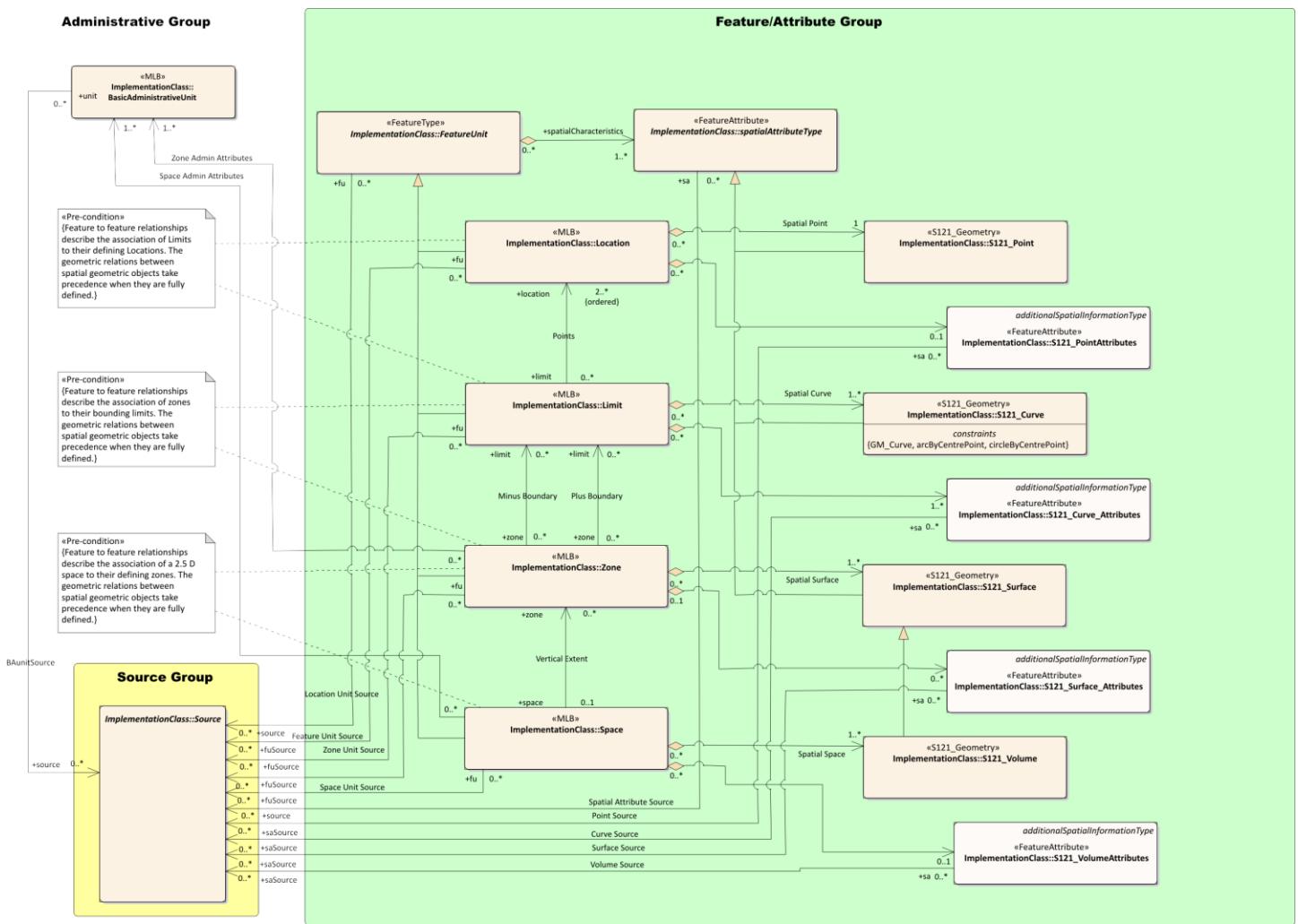


Figure 2-2 – S-121 Instantiated Implementation Model for Feature and Attribute Group

Figure 2-3 below shows the instantiable classes in the Feature/Attribute Group without the references to the Administrative Group or the Source Group.

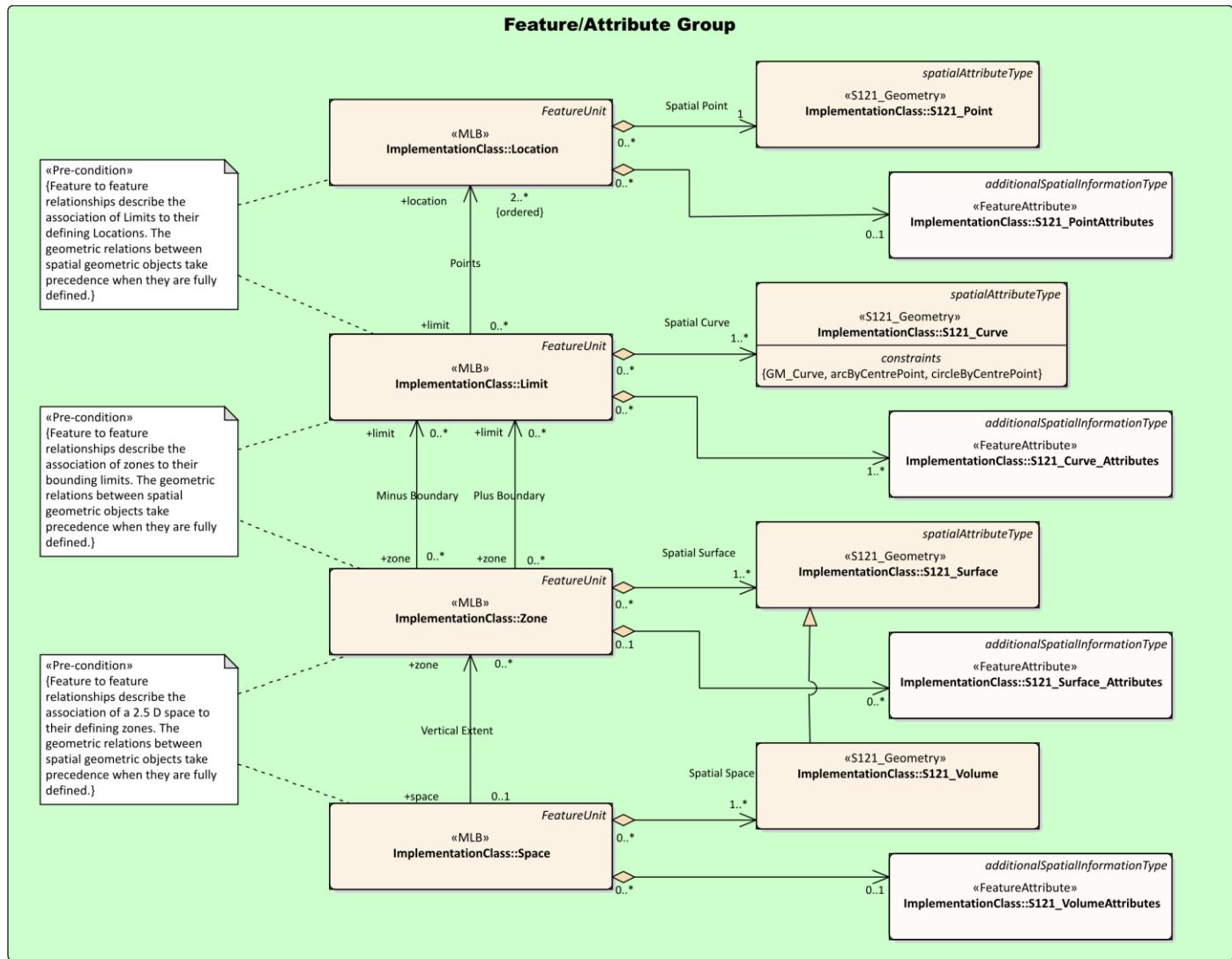


Figure 2-3 – S-121 Instantiated Implementation Model for Feature and Attribute Group without External References

Each of the classes has a set of defined attributes. These are shown separately so that they can be linked to the encoding.

Also the navigable relations shown in the figures above (Figures 2-1, 2-2 and 2-3) will need to be implemented as pointers in any encoding that uses them and will therefore appear as attributes (containing the Oid of the object pointed to). The use of pointers needs to be minimized in an explicit text encoding because it makes the text hard to read.

Figure 2-1 shows the Feature Unit. The Feature Unit will contain the name of the feature taken from Feature Catalogue. That is, it is a template for features such as “Territorial Sea”. Note that all of the attributes are character strings or code list entries that can be put in character string fields.

Some of the attributes, such as DateTime, are structured text. A date would look like yyyy-mm-dd hh:mm:ss:timezone⁵. Parts of the date are optional. This is described in more detail in the explicit text encoding section below.

⁵ The structure for this data format with year first is defined in ISO 19108 Temporal Schema [7] and ISO 8601 Date and Time Format [5]

The details of the implementation schema are given in Appendix A.

3 Explicit Text Encoding Format

An Explicit Text Encoding needs to be very simple and easily readable by a non-technical person. The format also needs to be parseable so that it can be read into a computer system.

The entire S-121 model allows for management of Maritime Limit and Boundary data as well as its expression in an exchange format. As such it contains capabilities, inherited from the ISO 19152 Land Administrative Domain Model, that are inherently complex such as per element versioning. Only a simple subset of the information contained in an S-121 model compliant database needs to be expressed through an explicit text exchange format interface.

The approach to the establishment of an Explicit Text Format used in this document is to structure text. This is illustrated in Figure 3-1. Structured text encoding goes back in the history of computing to the early delimited formats that were suitable for record oriented media. Such formats are simple and easily readable. In early computer systems this simple approach was necessary because of the limitations of the early systems; however, a delimited record oriented approach has an advantage because it is very easy to understand. Each piece of information forms a record and delimiters identify fields within the records. Records can be assembled into larger structures, which in this case are called blocks and groups. The information structure needs to be flexible and non-limiting, and the data fields need to be of variable length.

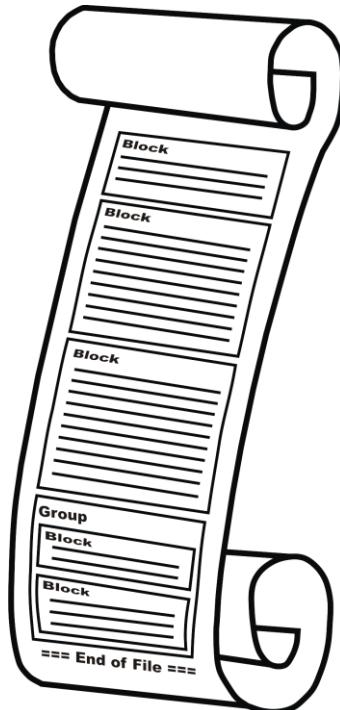


Figure 3-1 – Record, Group and Block Structure

3.1 Explicit Text Encoding Structure

There are three aspects to the Explicit Text Encoding Format. The first is the description of the delimiting structure that allows groups, blocks, records and sub records to be identified. This is described in clauses 3.3 to 3.3.4 below.

This structure is described in more detail in Appendix E using the Abstract Syntax Notation (ASN.1). ASN.1 is a widely used⁶ method for describing a syntax. ASN.1 has been standardized by ISO as standard ISO 8824.

Note: ASN.1, as used in this document, IS NOT AN ENCODING. It is a formal method of describing the Explicit Text Encoding.

The second aspect is the description of the contents of each block.

The third aspect of the Explicit Text Encoding Format is a profile that defines which elements are required and which are optional for a particular usage. The profile also establishes the text strings used as Block Descriptors and Record Identifiers. The profile given in Appendix D is for deposit with the UN.

Spatial information needs to be explicit. Coordinates need to be described as readable numbers; and lists of coordinates need to be lists of readable numbers.

The resultant Explicit Text Format needs to be printable with all of the information clearly available in the printed form. This means that there should be no hidden or “secret” unprintable characters⁷.

3.2 Cross referencing

Some information such as source statements, parties, rights, restrictions and responsibilities can be reused by a number of information objects. This is one of the powerful features of the S-121 model. However, cross referencing is difficult to follow in printed text and should be used sparingly. There are a few cases where it may be required. Where there is cross referencing between fields in the format each record that can be referenced needs to be uniquely identified, so that fields in one record can point to other records.

General numeric pointers may be conceptually simple but they are very difficult for a person to follow. A naming structure is needed for labels to make the pointers easy to follow. There are two ways that this can be accomplished, both of which are valid and may be used in the same data set. A pointer may make use of a unique identifier (unique ID) or a name. A unique ID is a structured object ID (Oid). Such an id consists of a block type name followed by a dash character delimiter (IA5⁸ Code 45) followed by a number. An example would be “Responsibility-1”. A name is an attribute value for a name field that is unique and can be used as an id. An example would be the use of the name “Canada” to refer to the party Canada described in a party type information object. Names are easier for humans to follow but they need to be managed to ensure that they are unique and meaningful. The use of names is preferred over the use of structured unique identifiers (Oid). Structured unique identifiers are easier to manage especially in the case when there are multiple references. For example “Source-12” uniquely identifies a source from a list of sources represented by a number of source information objects describing each source. Where there are multiple items of the same type names can sometimes become confusing. Both structured object ID (Oid) identifiers and names can be used in the same data set for different types of objects.

In practise it is best to use names where they are meaningful. Names work well for the Party, Right, Restriction, Responsibility and Basic Administrative Unit information object. Source and feature and geometry elements work best with Oid pointers. For example, an attribute for a Right information object might point to a right information object that was named “xxx”. The information object named “xxx” may provide details on the right. The pointer would read as:

Right: xxx

This type of “pointer” is very readable because it provides the information needed in the pointer.

⁶ There are many other schema languages for encoding formats that are now more popular than ASN.1; however, ASN.1 is well suited to an explicit text encoding, whereas schema languages such as XML Schema Definition Language (XSD) are suited to a delimited structure such as XML.

⁷ Some standards such as ISO/IEC 8211:1994 Information technology -- Specification for a data descriptive file for information interchange make use of imbedded control characters as delimiters making the plain text of the data very difficult to read.

⁸ IA5 is the International Alphabet number 5 standardized as ISO standard 646 International Reference Version and as a UN International Telecommunications Union recommendation T.50. It is equivalent to the US American Standard Code for Information Interchange (ASCII) except for the currency symbol \$ which may be substituted in some nations for a different currency symbol. IA5 is also the base of the UTF-8 character set used widely on the internet and the base page of Unicode (ISO 10646) multilingual character set.

3.3 Delimiting Structure

Computer systems can easily print data, but for it to be possible for that data to be read back into a computer it is necessary for the computer to know where each record of information starts and ends; and to be able to identify the record of information so that it can be placed in the correct place in the computer's internal information schema for S-121.

The S-121 Explicit Text Format makes use of five levels of structure. These are a Field (within a record), a Record, a Block and a Group (of blocks) which together compose a File.

The most important delimiter in a record oriented structure is the end of record delimiter. The earliest record oriented encoding format used punch cards where the length of a card was a fixed 80 characters. Data communications systems soon emerged after the era of the punch card and allowed variable length records. This required the use of a terminator to delimit the end of a record. Appendix B gives some background on why the sequence of CR (IA5 Code 13) followed by Line Feed LF (IA5 Code 10) character combination (CR LF) became the almost universal delimiter for an unstructured character string. Other systems that support CR or LF alone generally also support the CR LF combination. The use of the CR LF unstructured character string delimiter allows the S-121 Explicit Text Format to be read by almost all text editors and directly printed. The read operations in many languages such as Python may alter the data that is read and put it into the CR LF format. It is best to consider that CR alone, LF alone and CR LF are equivalent.

A record can be thought of as a “paragraph” in a printed document terminated by the CR LF (Carriage Return Line Feed) sequence. However, some attribute types allow unstructured text which may include vertical line spacing within the text. That is, an attribute value may contain several paragraphs. This needs to be allowed. There needs to be a clear way to distinguish the content of an attribute value composed of several records. For example a description attribute from a governance object may include several paragraphs of text that form one attribute value.

Description: Whereas Canada has long maintained and exercised sovereignty over the waters of the Canadian Arctic archipelago.

Therefore, Her Excellency the Governor General in Council, on the recommendation of the Secretary of State for External Affairs, pursuant to subsection 5(1)^{Footnote *} of the Territorial Sea and Fishing Zones Act, is pleased hereby to make the annexed Order respecting geographical coordinates of points from which baselines may be determined, effective January 1, 1986.

If one wishes to be able to write and then read such records it is necessary to be able to know where such a record begins and ends.

The intent of the S-121 Explicit Text Format is to allow any user to read, print and interpret a set of S-121 data without the need for special tools. Text processing software such as the simple “Microsoft Notebook” program or a word processor such as “Microsoft Word” or equivalent can easily read text delimited by a CR LF. Explicit text can be output in either a .txt⁹ or .rtf¹⁰ form. Text in a .txt form is simple text with no formatting. Text in .rtf form allows some formatting that can make the output easier to read, but the essential delimiting structure is the same. That is, .rtf encoded data can be converted to .txt encoded data with no loss of the S-121 encoded data, just a loss in the formatting.

Note that the .rtf format allows for the insertion of pictures, externally linked tables and other OLE¹¹ embedded items. Only a very limited use of OLE items can be allowed in an S-121 Explicit Text Format data file, and only in a few general freeform character string data fields. When converted to .txt format all imbedded OLE items will be discarded. The only valid use of imbedding an OLE item would be to assist in formatting the .rtf version of an Explicit Text Format S-121 data file. For example a picture of a logo may

⁹ .txt is a file type supported by Microsoft Word and many other text editors that consists of only IA5 or UTF8 characters with no imbedded formatting. It is the simplest type of file that exists on a computer system.

¹⁰ .rtf (Rich Text Format) is a Microsoft Word proprietary file format that imbeds some formatting in an otherwise simple text file. There have been many versions of .rtf since it was introduced in 1987. Microsoft stopped supporting .rtf in 2008 although current word processors still support the 2008 version.

¹¹ OLE - Microsoft Object Linking and Embedding

be inserted in character string text of the Title or Description attributes in the Governance object block. Such a picture of a logo would be deleted when the data is converted to **.txt** format or read back into a computer system. It is only there to beautify the printed form of the output.

An example of a title that includes an OLE object is shown below. This is a value field for a Title attribute in a Governance object represented in **.rtf** format. The imbedded image of a crest and the centered and bold formatting is entirely to enhance readability of the data to make it look like a printed document. When converted to a **.txt** form the OLE object and the formatting will disappear.

Example in **.rtf** format:



Example in **.txt** format:

Title: Maritime Boundary Definitions

The S-121 Explicit Text Format is parsable. That means that the data can be unambiguously read by a computer program and parsed allowing for it to be decoded into the original data structure from which it was generated.

The S-121 Explicit Text Format is record oriented. A record or “paragraph” often corresponds to an attribute value, but sometimes one or more records may be concatenated to form an attribute value.

3.3.1 Record Structure

The record structure is illustrated in Figures 3-2a and 3-2b below. There are two types of records each terminated by a CR LF end of record marker. A basic record starts with a record type identifier such as an attribute name followed by a TAB (IA5 code 09) character and then a character string (which may contain other TAB characters) terminated by the CR LF end of record marker. An extension record begins with a TAB character and then the record contents. An extension record is an extension of the record before it. For example, an extension record may contain a second or subsequent paragraph in a text description that includes CR LF as part of the data.

Record



Figure 3-2a – Record Structure

Extension Record

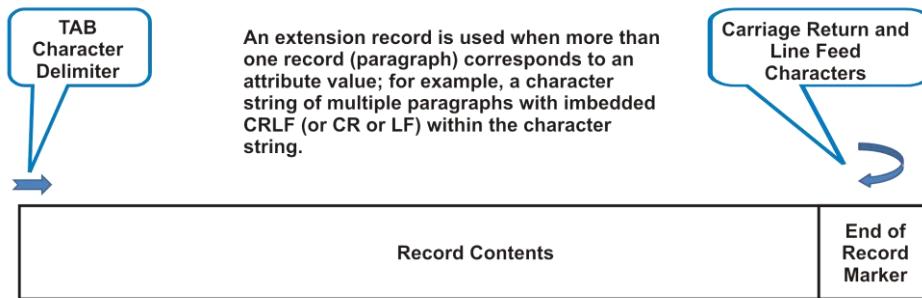


Figure 3-2b – Extension Record Structure

A record content may contain several data fields representing values separated by the COMMA character (IA5 code 44) and an optional SPACE character (IA5 code 32). This is shown in Figure 3-3.

Record Contents

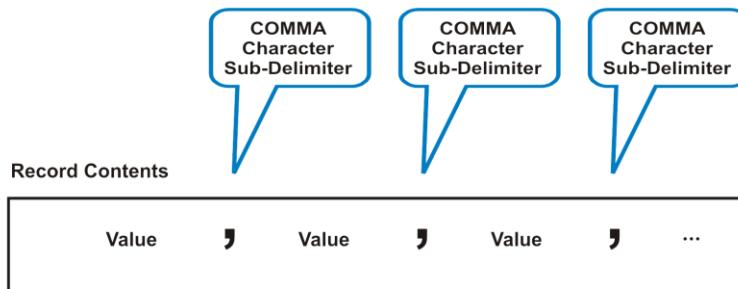


Figure 3-3 – Record contents with delimited sub-fields

An example of a simple record would consist of an attribute name followed by a value.

Title: Test data set 1

An example of a record with multiple values would be an attribute name followed by a list of values separated by commas.

Topic: boundaries, oceans

An example of a multi-line record that includes CRLF in its value field is shown below. This consists of an attribute name followed by a record contents and additional contents in an extension record.

Description: This is an example of a textual data that consists of multiple lines of text which automatically wraps when printed.

This text also includes a second paragraph which is still part of the attribute value field, but there is a CR LF character within the text. This second paragraph of text makes use of an extension record and as such begins with a TAB character causing the second paragraph of text to be indented.

3.3.2 Data Blocks

The Explicit Text Encoding Format makes use of a set of independent data blocks. Each block corresponds to a different type of data element from the S-121 Implementation Model. Each block contains a number of records and each record is terminated by a CR LF. Each block begins with a Block Descriptor Record from

which one can determine the type of block. The Block Descriptor Record may be a string of text that is meaningful in informing the reader of the purpose of the Data Block. All that is necessary is that it is unique. The Block Descriptor Record serves two purposes. It is a unique identifier for the block and also a header to inform the reader of the meaning of the block. The exact string used is defined in the profile. Each block ends with a double CR LF CR LF (double blank line).

The content that is expressed in the Explicit Text Encoding Format is deliberately simplified from the full capability of the S-121 schema. That is, more complex data may be held in a database than can be expressed in a single output data file.

The first data block is unique. It always corresponds to Metadata. The second block always corresponds to the Governance element. The Basic Administrative Unit block may be anywhere in the dataset and is identified by the Block Descriptor Record. There may only be one Governance block and one Basic Administrative Unit block per dataset. If a usage requires more than one Governance element or more than one Basic Administrative Unit it should be output as more than one data file in the Explicit Text Format.

Figure 3-4 below illustrates a block. The first record is a Block Descriptor Record followed by any number of other records and terminated by CR LF CR LF.

Block Structure

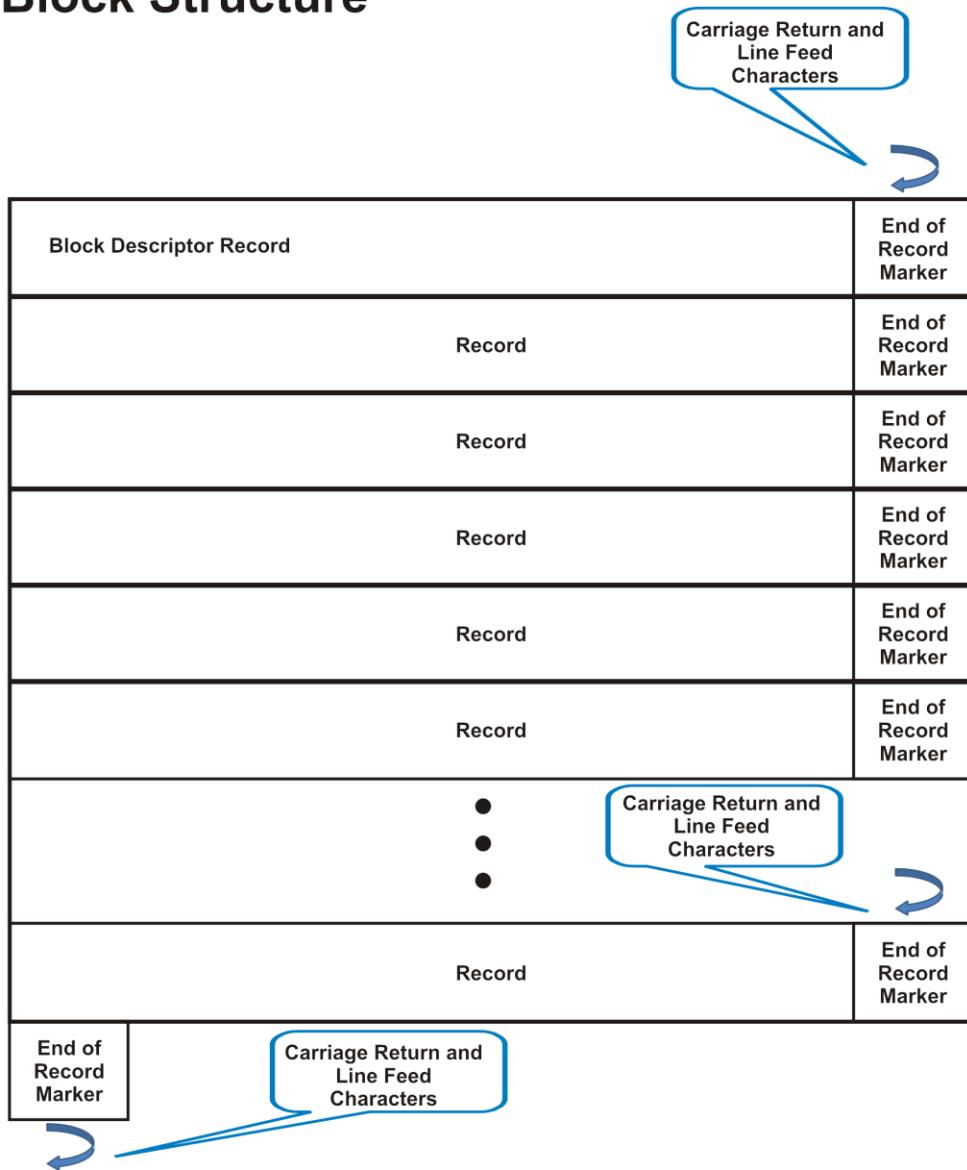


Figure 3-4 – Block Structure

3.3.3 Data File

A data file is composed of a set of data blocks terminated by an end of file indicator. The end of file indicator is a data block that contains a single record that is a specific character string. This string is: === End of File === . Since the end of file is a data block it has both the end of record terminator CR LF and the end of block CR LF. That is, an end of file is the string === End of File ===CRLFCRLF. This is illustrated in Figure 3-5 below.

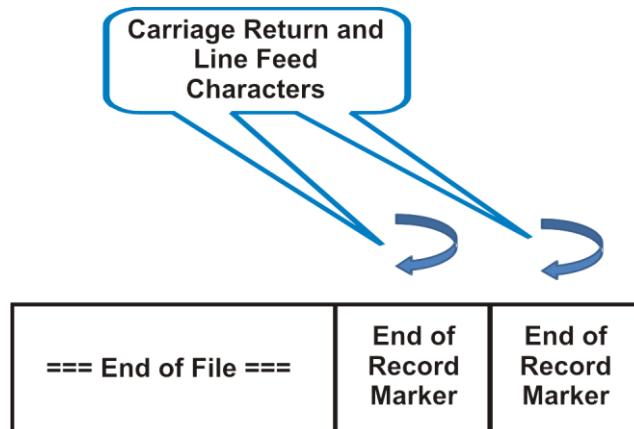
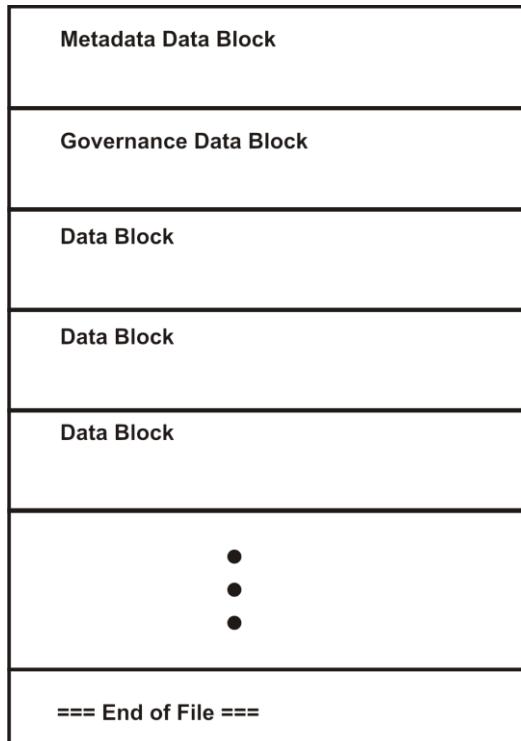
***Figure 3-5 – End of File***

Figure 3-6 below presents a data file consisting of a set of data blocks. The first data block is Metadata followed by the Governance data block. After that there may be one Basic Administrative Unit block and any number of feature blocks, spatial element blocks and other element blocks, in any order, terminated by an end of file indicator.

For simplicity there may only be one Metadata, one Governance and one Basic Administrative Unit block per dataset. In situations where multiple Governance or Basic Administrative Unit elements needs to be described, multiple data files need to be used to express the data in the S-121 Explicit Text Format.

***Figure 3-6 – Dataset File Structure***

3.3.4 Block Descriptor Record

The block descriptor record uniquely identifies the type of data block. The identification occurs in the first field of the first record in the data block. This is the record “type identifier” field. For some block types only the type identifier is necessary.

For some block types, such as a “Source” block type it may be necessary to reference this block from attribute records in other blocks. In this case the second field of the block descriptor record would contain a unique identifier.

Pointers and identifiers are hard to follow for human readers. They should be used sparingly. The structure of a Block Descriptor Record is illustrated in Figure 3-7 below.

Simple Block Descriptor Record



Block Descriptor Record with ID

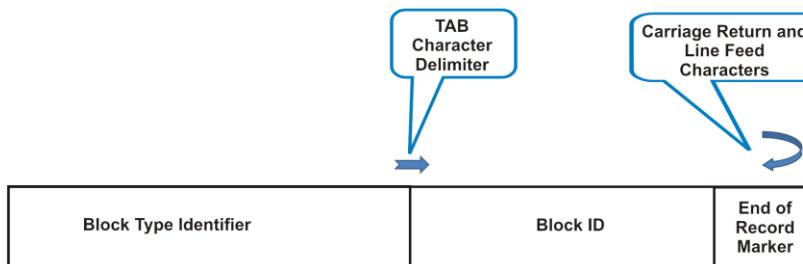


Figure 3-7 – Block Descriptor Record

The Block Type Identifier may be a character string of any (reasonable) length. It could be a whole sentence or two that describes the purpose of the data block to the reader.

An example of a Block Type Identifier is given below:

Metadata:

Note that the Block ID needs to be a unique character string. Commonly that would be a type of an element followed by a unique number (an Oid). For example an identifier could be “Right-1”. The structure of the Oid identifier is shown in Figure 3-8 below.

Object Identifier (Oid)

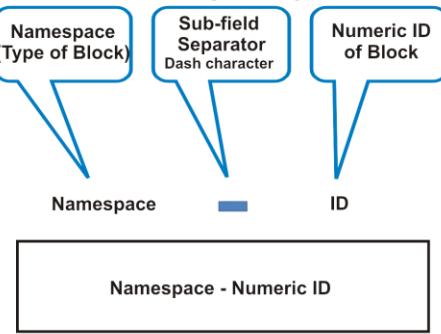


Figure 3-8 – Object Identifier

Table 3-1 – Namespaces for use in the object identifier

Rights Namespace	"Right-"
Restrictions Namespace	"Restriction-"
Responsibilities Namespace	"Responsibility-"
Party Namespace	"Party-"
Source Namespace	"Source-"
Location Namespace	"Location-"
Limit Namespace	"Limit-"
Zone Namespace	"Zone-"
Space Namespace	"Space-"
Point Namespace	"Point-"
Curve Namespace	"Curve-"
Surface Namespace	"Surface-"
Space Namespace	"Space-"

The record NumericID is a unique number in the namespace for the record instance. In the Explicit Text Format this is represented as a character string containing numeric characters. For example a party may be identified as "Party-3".

A Block ID may also be a "name". Names may be used to make the data more readable. For example the Block ID for a Party type of information block may be "Australia". This means that a reference to that information block party would be the name "Australia". In this case the name "Australia" would appear as an attribute value in the reference so that the reader would immediately understand the reference rather than having to follow the reference to determine that "Party-1" was Australia. Either unique names or structured Object Identifiers may be used as identifiers and therefore included in the Block ID field.

A Block Descriptor Record may also be extended using an extension record. That is, an extension record may immediately follow a block descriptor record. This extension record adds to the description, but it does not extend either the Block Type Identifier field or the Block ID field. The extension allows for additional text to be added to increase human readability of the data. An example of an extended Block Descriptor Record is shown below. The Block Type Identifier field is "Maritime Limits and Boundaries Deposit" establishing this as a Governance object block. In this case there is no Block ID. The extension contains additional description.

Maritime Limits and Boundaries Deposit:

This deposit addresses the arctic waters in Canada's region 7.

3.3.5 Data Block Groups

A set of data blocks of the same type can form a group. For example, two parties may be defined in a particular data set. A Party Data Block Group may be established that begins with descriptive text that identifies the set of Party Data Blocks, followed by the definitions of the individual blocks.

Since a Spatial Attribute Type object (Location, Limit, Zone, Space) is related to a Feature Unit object (Point, Curve, Surface, Volume) by a composition relationship the two block types can be included in the same group. This allows for the geometry to immediately follow the Feature Unit object to which it is related. This ordering and grouping of blocks is not mandatory, but it enhances readability.

A data block group begins with a data block that consists of only a Block Descriptor Record. This allows the introductory text to be expressed. The Block Type Identifier of the Block Descriptor Record needs to be unique to identify the type of Group.

Technically Data Block Groups are not required because each data block is individually identified, but the use of groups allows for additional Block Descriptor Records to be expressed that makes the text more readable.

Data Block Groups are not directly referenced so the Block ID field / Group ID field (from the Descriptor Record) remains blank.

The Block Descriptor Record that establishes a Data Block Group may also be extended using an extension record. This extension record adds to the description to allow for additional text to be added increasing human readability of the data. The Block Type Identifier field (that establishes the Group) is not extended.

Attributes that are common to all of the information objects in a group may be expressed as attribute records immediately following the Data Block Group header record. This allows certain attributes such as a Source, or a Coordinate Reference System that may be constant for all information objects in a group to be stated once. This reduces the verbosity of the format.

An example is given below that shows a Data Block Group composed of several Zone Blocks.

The extent of the object is defined by the following zones:

Zone Identifier: **Zone-1**

...

Zone Identifier: **Zone-2**

...

3.3.6 Table Structure

A complement to the record structure is the table structure. A table is a special type of block. A table is an efficient and readable way to present a large number of instances of data elements. A data block presents one information object with its attributes. A table presents many information elements of the same type together with their attributes. A single information element corresponds to a row and attributes are represented in columns. A table is equivalent to a set of data blocks. The restriction is that the table can only be so wide. A table works well if there are only a few attributes. The limitation is the width of a printed page in an Explicit Text Format that is meant to be read by humans.

The first row of a table is the BlockDescriptorRecord. This identifies the type of information element being represented by the table. The BlockDescriptorRecord may also contain a BlockID. The BlockDescriptorRecord is described in clause 3.3.4 above.

The second row of the table contains a list of Record Type Identifiers separated by TAB characters. This identifies the attribute types corresponding to each column.

The third and subsequent rows contain values corresponding to each table column.

The table is terminated by an end of block sequence which is a double CR LF CR LF (double blank line).

A table structure is illustrated in Figure 3-9 below.

Table Structure

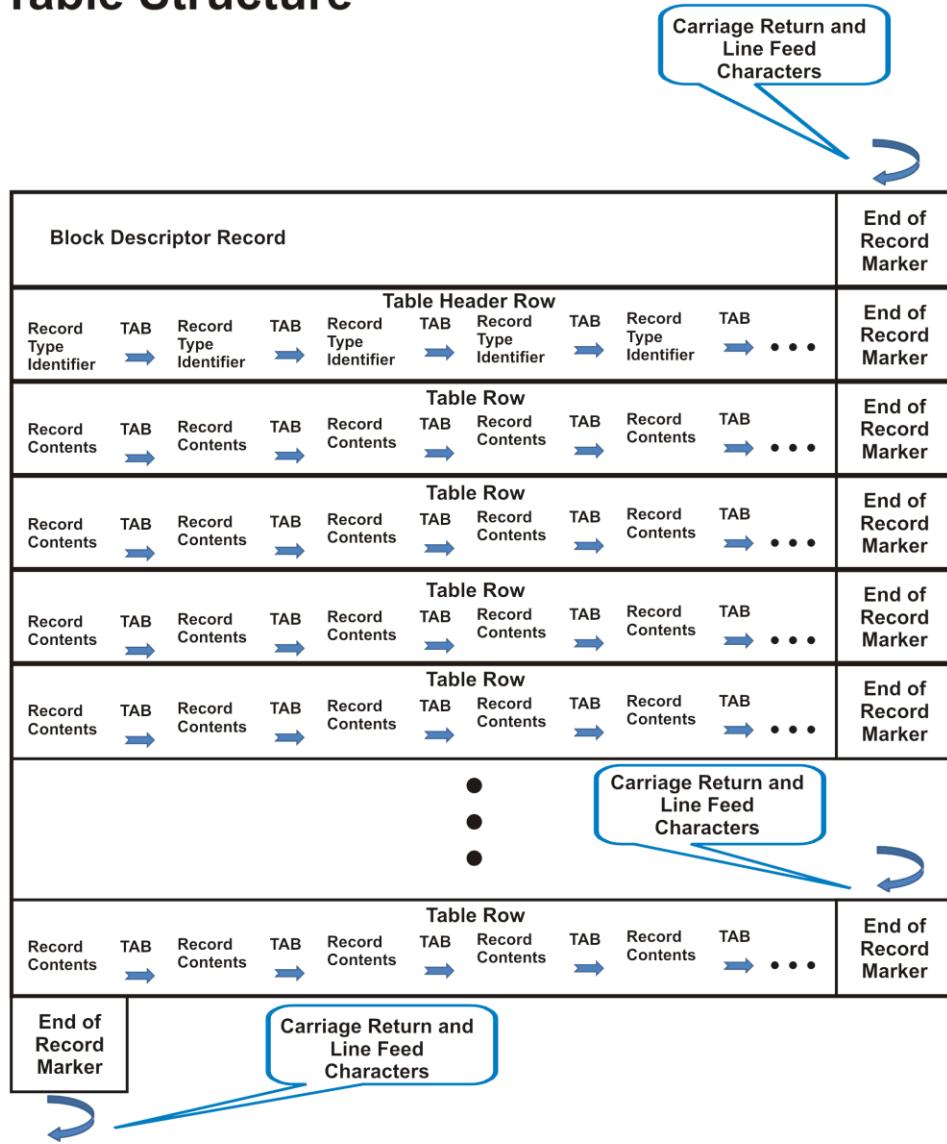


Figure 3-9 – Table Structure

3.4 Data Types

There are several data types used within the Explicit Text Format. These data types are represented as formatted text strings. These formatted strings are described below.

Character String – a string of IA5¹² or UTF8¹³ characters. The character set and the language are described in the metadata. Although a data base may be multi-lingual making use of structures such as the ISO 19115-3 LOCALE mechanism to imbed text strings with language codes, the Explicit Text Format is unilingual. To address the same data in two different languages, two different data sets should be issued.

¹² International Alphabet 5 from ISO 646 -- equivalent to Basic ASCII text

¹³ Unicode Transformation Format 8 bit from ISO 10646 – extended UNICODE character set.

Coordinate - A number of coordinate formats are permitted; however, it is recommended that only one coordinate format be used for one type of coordinate in a dataset. Indicators of East / West and North / South may be either -- E / W and N / S or + / -. When + and – are used the + is optional. The delimiters for degree, minutes and seconds of the symbols ° (U+00B0)¹⁴, ' (U+2032), and " (U+2033) may be used.

Table 3-2 – Coordinate data types

DegreeMinuteSecondDecimalSecondsCardinal	(N/S or E/W) DD MM SS.S
DegreeMinuteDecimalMinutesCardinal	(N/S or E/W) DD MM.M
DegreeDecimalDegreeCardinal	(N/S or E/W) DD.D
DegreeMinuteSecondDecimalSecondsSign	(+ or -) DD MM SS.S
DegreeMinuteDecimalMinutesSign	(+ or -) DD MM.M
DegreeDecimalDegreeSign	(+ or -) DD.D

Coordinates follow the structure defined in the ISO standard 6709¹⁵. ISO defines the order Latitude then Longitude.

Coordinates may be expressed in many different formats in national legislation and treaties and it is important that the exact representation in those primary sources be retained. For this reason there is a lot of flexibility in the Explicit Text Format on how to represent coordinates.

Direct Position – A direct position (Latitude, Longitude) position will be represented as a sequence of coordinates separated by a delimiter sequence SPACE Slash SPACE “ / ” (IA5 codes 32, 47, 32).

DateTime – Date and Time are represented in a structured character string. The format is based on the standards ISO 8601 and ISO 19108.

The format is YYYY:MM:DD:hh:mm:ss:timeZone

The sub fields are separated by the delimiter : (IA5 code 58)

YYYY is the year; for example 2017

MM is the month; for example 05 (May)

DD is the day; for example 17

hh is the hour; for example 13 (in 24 hour clock)

mm is the minutes; for example 15

ss is the seconds; for example 00

timeZone is a shift from Greenwich Mean Time in hours; for example -5 for Eastern North America

The plus sign + is optional for east of Greenwich.

Any unknown fields may be represented by consecutive delimiters; for example ::

Least significant fields and delimiters may be omitted.

An example date may be 2018:05:12

An example date/time would be 2017:05:17:13:15:00:-5

Fraction - a data type corresponding to only the fractional part of a number will be represented as the text string "0.xxxx..." where xxx... is the mantissa (fractional part of a number).

Boolean - a Boolean data type will be represented as the text “Yes” or “No”.

3.5 Information Elements

The S-121 model consists of a set of information elements described by classes in the S-121 UML model. The implementation model is presented in Figures 2-1 to 2-3. The classes are organized into four groups,

¹⁴ (U+xxxx) represents a ISO 10646 Unicode UTF8 code.

¹⁵ ISO 6709 Standard representation of geographic point location by coordinates

the Feature/Attribute Group; the Administrative Group; the Party Group; and the Source Group. The Feature/Attribute group contains eight classes, Location; Limit; Zone and Space with Point; Curve; Surface; and Volume as associated component classes. The Administrative group contains five classes, Governance; BasicAdministrativeUnit; Rights; Responsibilities; and Restrictions. The party group contains two classes, Party; and Party Member. The Source group contains one class, Source. Together this is sixteen classes. Each class is represented by a block type. Together with metadata this is seventeen block types. The following subsections list each block type together with the allowed attributes and their definitions.

These block types correspond to the instantiable types from the Implementation Model. Each type may have a different structure and has a different namespace for the Record Number.

- Metadata Block
- Governance Block
- Basic Administrative Unit Information Block
- Rights Block
- Restrictions Block
- Responsibilities Block
- Party Block
- Party Member Block
- Source Block
- Location Block
- Limit Block
- Zone Block
- Point Spatial Attribute Block
- Curve Spatial Attribute Block
- Surface Spatial Attribute Block

The conditionality of whether an attribute or information object is Mandatory or Optional is derived from the model.

The attribute name column in the tables provides a name for the attribute that can be output as text in the "Record Type Identifier" field. Alternatively, an alias or alternate name can be defined in a profile that may be more meaningful to the human reader for that particular profile.

3.5.1 Metadata Block

Metadata is required in order to discover and to interpret data and is needed for data exchange.

In the Explicit Text Format may be expressed, but it is not required. The metadata used in a legislative instrument is implicitly included in a governance object. When used the metadata block is the first data block in a dataset. If metadata is not expressed an empty metadata block is required. That is, if no metadata is expressed then a dataset must begin with an end of block terminator (two consecutive end of record terminators, which is CR LF CR LF (or CR CR or LF LF).

The metadata fields are described in clause 1.2.2. The details of which metadata fields and which Record Type Identifiers are used to identify the metadata fields for a particular situation is described in a profile.

Note that when used, metadata forms a cover page for Explicit Text Format data submitted to UN/DOALOS.

Table 3-3 – Metadata Block - optional

Attribute Name	Description
title	Name by which the cited resource is known
abstract	Brief narrative summary of the dataset's contents
topic	The main theme(s) of the dataset from the ISO list of topic areas: farming, biota, boundaries , climatology, Meteorology, Atmosphere, economy, elevation, environment, geoscientificInformation, health, imageryBaseMapsEarthCover, intelligenceMilitary, inlandWaters, location, oceans , planningCadastre, society, structure, transportation, utilitiesCommunication
metadataContactName	Name of the individual responsible for the metadata
metadataContactOrganization	Organization responsible for the metadata
metadataContactPosition	Position of the individual responsible for the metadata
metadataContactRole	Role of the individual responsible for the metadata
referenceDate	Reference date for the cited resource
datasetLanguage	Languages of the dataset using standard ISO 639-2 three letter codes. Note: Three letter language code followed by an optional ISO 3166 three letter country code, For example ENG for English and FRA for French
metadataDate	Metadata creation date
metadataFileIdentifier	A unique phrase or string which uniquely identifies the metadata file
metadataLanguage	Language of the metadata composed of an ISO 639- 2/T three letter language code and an ISO 3166-1 three letter country code
metadataCharacterSet	character coding standard in the metadata. Default UTF8
metadataFileParentIdentifier	The unique name of the file or associated fileIdentifier, related in higher hierarchy to the file
metadataHierarchyLevel	Level to which the metadata applies. (for example dataset)
metadataHierarchyLevelName	Name of the level to which the metadata applies. (for example dataset)
datasetReferenceDateType	Type of date from code list <ul style="list-style-type: none">- creation date- publication date- revision date- date became not available- date became in force- date became adopted- date became deprecated- date became superseded
datasetCharacterSet	Character coding standard in the dataset. Default UTF8
westLongitudeExtent	West limit of extent
eastLongitudeExtent	East limit of extent
southLatitudeExtent	South limit of extent
northLatitudeExtent	North limit of extent

3.5.2 Governance Block

The second block of data in the S-121 Explicit Text Format is the Governance Block. The S-121 Governance objects carry the high level information that describe the context of the dataset. Governance information is a specialized type of metadata that is in excess of the metadata defined to allow discovery and interpretation of a dataset. The governance information is the preamble to the “list of points” that forms the body of the data submitted. It is often the title, reference number and preamble text to a set of data.

In the S-121 schema multiple governance objects and multiple Basic Administrative Unit objects are permitted. This is in the database holding Marine Limit and Boundary data. For simplicity only one Metadata Block, one Governance Block and one Basic Administrative Unit (BAUnit) Block are permitted in the explicit exchange format.

A Governance Block is optional. If a Governance Block is not expressed an empty block is required with an end of block terminator.

Since the Governance Block is always the second block a Block Type Identifier (BlockDescriptorRecord) is not required to determine which block is the governance block. The Block Descriptor Record for the governance block may or may not be expressed. If it is not expressed then a blank record is required terminated by an EndOfRecordMarker.

The content of a Governance Block is also structured. All of the attributes are optional or conditional. Which ones are used is described in a profile. The StartLifespan (Version) attribute may duplicate the Reference Date attribute in metadata and is not required when this metadata attribute is expressed.

The following Table outlines the attributes of the Governance class in the S-121 schema and how they map to a Governance Block in the Explicit Text Format. The first column is the attribute name in the S-121 schema. The second column is a description of the information carried in that attribute.

Table 3-4 – Governance Block - optional

Attribute Name	Description
label	A short textual identifier of the governance object
referenceNumber	The reference number of the legislative reference. This would be the reference number used by the nation for its proclamation, law or other legislative document
name	The name of the governance object. When used, this name is defined by the nation state
governanceTitle	Title of the official reference. This is the general title of the whole deposit
governanceDescription	Description of the governance statement. This can be a multi paragraph preamble to the deposit
releasabilityType	Releasability status. From the code list (official, internal, controlled)
dateApproved	Date at which the official statement or document was approved by the appropriate governing body
startLifespan	Reference date for the validity of this version of the dataset. This is the statement of versioning for an entire dataset. Note that the Explicit Text Format is versioned by dataset. If Metadata is provided then this date is equivalent to Reference Date
endLifeSpan DEFAULT ("Current")	End date of a specific instance version. This is the complementary attribute to the StartLifespan attribute. However, since the Explicit Text Format is versioned by dataset it is rare that this attribute would contain any value other than the default “current” so it would normally not be expressed
dateConsidered	The date at which the official statement or document was considered by the appropriate governing body
dateIntroduced	The date at which the official statement or document was introduced
collection	Description of the feature membership to a dataset or collection. For example, a set of data is part of a particular nation’s collection; that is, was deposited by that nation. It is expected that the State name should be used when depositing

3.5.3 Basic Administrative Unit Block

The central element of the S-121 model is the Basic Administrative Unit element. It is the interface point into the administrative structure of S121 Marine Limit and Boundary data. It contains references to Party and Source attributes by reference and Rights, Restrictions, Responsibility attributes. In the general S-121 schema the Rights, Restrictions, and Responsibility attributes are also handled as attributes by reference; however to simplify the Explicit Text Format the Rights, Restrictions, and Responsibility attribute information, maybe collapsed into simple attributes of the Basic Administrative Unit.

The Basic Administrative Unit object in the S-121 schema allows multiple objects to be defined. This allows the data in a S-121 database to be organized into logical units. For simplicity the Explicit Text Format only allows one Basic Administrative Unit to be expressed at one time.

In the S-121 schema the Basic Administrative Unit object(s) may be dynamically versioned. This is one of the important capabilities of S-121. However, in the Explicit Text Format versioning is done at the dataset level based on the reference data in the Metadata or in the Governance object. Therefore the startLifeSpan, endLifespan and collection attributes described in the S-121 model do not appear in a Basic Administrative Unit group in the Explicit Text Format.

Table 3-5 – BAUnitBlock - optional

Attribute Name	Description
basicAdministrativeUnitName	Optional name for an instance of a "BasicAdministrativeUnit"
basicAdministrativeUnitType	The use type of a "BasicAdministrativeUnit" = MaritimeLimitsAndBoundaries. (Fixed string, therefore not required)
basicAdministrativeUnitContext	Description of the context for an instance of a "BasicAdministrativeUnit"
rights	Reference to Right object
responsibilities	Reference to Responsibility object
restrictions	Reference to Restriction object
party	Reference to Party object
source	Reference to Source object

3.5.4 Rights Restrictions and Responsibilities Blocks

The Right, Restriction and Responsibility objects carry the descriptions of the right, restriction and responsibility attributes. These are shared attributes through referencing.

3.5.4.1 Rights Block

An S121_Right is an action, activity or class of actions that a system participant may perform on or using an associated resource. The RightsBlock describes a right.

Table 3-6 – RightsBlock - optional

Attribute Name	Description
rID	Identifier of the Rights block in the Oid format or is a "name" for the Rights block
rightType	The type of the right
description	Description of the right
share	Share of an instance of a right
shareCheck	Boolean indicating whether the constraint is applicable
timeSpecification	Operational use of a right in time sharing
party	Reference to Party object
source	Reference to Source object

3.5.4.2 Restrictions Block

An S121_Restriction is a formal or informal entitlement to refrain from doing something. The RestrictionBlock describes a restriction.

Table 3-7 – RestrictionBlock - optional

Attribute Name	Description
rID	Identifier of the Restrictions block in the Oid format or is a “name” for the restriction block
restrictionType	The type of the Restriction
partyRequired	Indicates whether a party is required for the registration of the restriction in the association to Party
description	Description of the restriction
share	Share of an instance of a restriction
shareCheck	Boolean indicating whether the constraint is applicable
timeSpecification	Operational use of a restriction in time sharing
party	Reference to Party object
source	Reference to Source object

3.5.4.3 Responsibilities Block

An S121_Responsibility is a formal or informal obligation to do something. The ResponsibilityBlock describes a responsibility.

Table 3-8 – ResponsibilityBlock - optional

Attribute Name	Description
rID	Identifier of the Responsibilities block in the Oid format or is a “name” for the Responsibility block
responsibilityType	The type of the Responsibility
description	Description of the responsibility
share	Share of an instance of a responsibility
shareCheck	Boolean indicating whether the constraint is applicable
timeSpecification	Operational use of a responsibility in time sharing
party	Reference to Party object
source	Reference to Source object

3.5.5 Party Group Blocks

A party is a person or organisation or group that plays a role in a rights transaction. A group is a type of party that has other parties as members. This means that there may be relations between parties that describe membership in a group. Party members may also have a share of a group. In the S-121 model this is expressed as an attribute on the membership relationship; however, in an implementation this has been flattened to be a second attribute in the party object.

3.5.5.1 Party Block

An S121_Party is a person or organisation or group that plays a role in a rights transaction. The PartyBlock describes a party. The party contains both a pID and a Party Name Party attribute.

Table 3-9 – PartyBlock- optional

Attribute Name	Description
pID	Identifier of the Party block in the Oid format or is a “name” for the Party block
partyName	The name of the party (Not needed is the “Name” is used as the pID)
partyRole	The role of the party
partyType	The type of the party <ul style="list-style-type: none"> - internationalOrganization - stateCountry - provinceState - nonNaturalPerson - naturalPerson
groupPartyType	The type of the group party: “agreement” or “association”
externalPartyID	The identifier of the party in an external registration
IsIndividualOrGroupParty	“individualParty” or “groupParty”
partyMemberIdentifier	Identifier of a Group Party in the Oid format
source	Reference to Source object

3.5.5.2 Party Member Block

A party may be a group and have members. A Party Block may be identified as a Group and have references to its members. A member may have a particular share of a group. The Party Member Block carries the attribute Share which indicates the share of a party.

Table 3-10 – PartyMemberBlock - optional

Attribute Name	Description
pID	Identifier of the Party Member Block in the Oid format or is a “name” for the Party member
rID	Reference to Party Block
Party Share	Fraction of the group represented by the party member

3.5.6 Source Block

All of the information elements in the S-121 model may be sourced. This is an important capability of the standard. The Source Block carries the information for a source. The source element includes a large number of optional elements that allow a source to be described. Many of these derive from the ISO metadata standards. The “name” of a source is not normally an alternative for the sID because source document names may be long and complex.

Table 3-11 – SourceBlock - optional

Attribute Name	Description
sID	Identifier of the Source block in the Oid format (or in rare cases is a “name” for the source)
sourceDocumentName	Document name - for example the document (legislation, treaty, title) that defines the object
sourceOnlineResourceLinkageURL	Official URL (or equivalent online resource) where the document is distributed; Multiple online resource URL may be included in one character string separated by commas
sourceOnlineResourceProtocol	Online Resource Protocol. Multiple online resource protocols corresponding to the multiple URLs may be included in one character string separated by commas

Attribute Name	Description
sourceOnlineResourceApplicationProfile	Name of an application profile that can be used with the resource. Multiple application protocols corresponding to the multiple URLs may be included in one character string separated by commas
sourceOnlineResourceName	Name of the online resource
sourceOnlineResourceDescription	Description of what the resource is/does
sourceOnlineResourceFunction	Function performed by the resource: download, information, offline access, order, search
sourceRegistryNumber	Unique official identifier of the record in a registry. For example, in states with registers of legislative instruments, versioning is controlled by the registry ID
sourceAdministrativeTimeStamp	The moment that the event represented by the instance of S121_Source is further processed
sourceAuthoritativeDate	The date of force of law of the source by an authority
sourceDocumentType	The type of document
sourceAvailabilityStatus	<p>The status of document from the code list LA_AvailabilityStatusType;</p> <ul style="list-style-type: none"> - archiveConverted - archiveDestroyed - archiveIncomplete - archiveUnknown - docAvailable <p>The default is "docAvailable" and the field does not need to be expressed when the source document is known to be available</p>
administrativeSourceType	Descriptive documentation that supports, complements or describes the associated object
spatialSourceType	The type of spatial "Source" document
sourceType	The type of "Source" document Reference
sourceSubmissionDate	The date of submission of the source by a party
sourceRecordation	The date of registration (recordation) of the "Source" by registering authority
responsiblePartyIndividualName	Name of the responsible individual
responsiblePartyContactOnline	Method, source, or location for online access
responsiblePartyOnlineProtocol	Connection protocol to be used
responsiblePartyOnlineApplicationProfile	Name of an application profile that can be used with the resource
responsiblePartyOnlineContactName	Name of the resource
responsiblePartyOnlineDescription	Function performed by the resource:
	<ul style="list-style-type: none"> - download - information - offlineAccess - order - search
responsiblePartyOrganizationName	Name of the responsible organization
responsiblePartyPositionName	Name of the role or position of the responsible person
responsiblePartyContactPhone	Telephone numbers at which the organization or individual may be contacted

Attribute Name	Description
responsiblePartyRole	Role of the responsible party from code list: - resourceProvider - custodian - owner - user - distributor - originator - pointOfContact - principleInvestigator - processor - publisher - author
responsiblePartyContactAddressCountry	Country of the physical address
responsiblePartyContactAddressDeliveryPoint	Address line for the physical address (street name, box number, suite)
responsiblePartyContactAddressCity	City of the physical address
responsiblePartyContactElectronicEmailAddress	Electronic mail address
responsiblePartyContactAddressAdministrativeArea	State, province of the physical address
responsiblePartyContactAddressPostalCode	Postal Code for the physical address
qualityNameOfMeasure	Name of quality measure. Multiple measure names may be included in one character string separated by commas
qualityMeasureIdentification	Identification of quality measure
qualityMeasureDescription	Description of quality measure
qualityMeasureEvaluationType	Type of quality evaluation method
qualityEvaluationMethodDescription	Description of quality evaluation method
qualityEvaluationProcedure	Procedure for measuring quality
qualityDateTime	Reference date and time for quality. Multiple measure data times corresponding to the multiple names may be included in one character string separated by commas
qualityResult	Result of data quality measure. A second result may be included one character string separated by a comma
externalArchiveAcceptance	The date of force of law of the source by the authority
externalArchiveData	The content of the source
externalArchiveExtraction	Date of extraction from external archive
externalArchiveExtraction	The date of registration (recordation) of the source by registering authority
externalArchive_sID	The identifier of the source
externalArchiveExtraction	The date of submission of the source by a party

3.5.6.1 Location Block

The Location block is an object that defines the underlying structure of location. Note that the Oid block ID is used. Name and Label are not alternatives for referencing.

Table 3-12 – LocationBlock - optional

Attribute Name	Description
fulD	Identifier of the Location object feature unit in the "Oid" (Object ID) format. The "Oid" comprises a unique character string and a namespace identifier which is also a unique character string. An example might be "Location-1" where "Location" is the namespace and "1" is the unique id within that namespace
label	A short textual identifier of the feature unit
name	The name of the feature
context	The administrative aspects of the feature object
releasabilityType	This attribute is optionally used to differentiate between releasability status for particular features
locationObjectType	This attribute identifies which Feature Unit Location Object type this object refers to based on the Feature Catalogue: - location - baselinePoint - contributingPoint - limitPoint - boundaryPoint
interpolationRole	The role of point in the structure of a "Straight" line or curve: - deflection - densification
pointType	The type of point
administrativeUnit	Reference to the Basic Administrative Unit to which this feature is associated
source	Reference to the source object
point	Reference to the associated point geometric object

3.5.6.2 Limit Block

The Limit block is an object that defines the underlying structure of Limit. Name and Label are not alternatives for referencing.

Table 3-13 – LimitBlock - optional

Attribute Name	Description
fulD	Identifier of the Location object feature unit in the "Oid" (Object ID) format
label	A short textual identifier of the feature unit
name	The name of the feature
context	The administrative aspects of the feature object
releasabilityType	This attribute is optionally used to differentiate between releasability status for particular features

Attribute Name	Description
limitObjectType	This attribute identifies which Feature Unit Limit type this object refers to based on the Feature Catalogue: <ul style="list-style-type: none"> - Boundary - International Boundary - Baseline - Normal Baseline - Straight Baselines, - Archipelagic Baseline - Low Tide Elevations - Mouths of Rivers - Reefs - Bays - Ports - Outer Limit of the Territorial Sea - Outer Limit of the Contiguous Zone - Outer Limit of the Exclusive Economic Zone - Outer Limit of the Continental Shelf - Outer Limit of the Roadstead - Construction Line
arcGeometryType	Type of computation used to define an arc (line): <ul style="list-style-type: none"> - geodesic - loxodrome
administrativeUnit	Reference to the Basic Administrative Unit to which this feature is associated.
source	Reference to the source object
curve	Reference to the associated curve geometric object
delimitingPoints	Reference to the beginning and end points that delimit a limit

3.5.6.3 Zone Block

The Zone block is an object that defines the underlying structure of zone. Name and Label are not alternatives for referencing.

Table 3-14 – LocationBlock - optional

Attribute Name	Description
fulD	Identifier of the Zone object feature unit in the "Oid" (Object ID) format
label	A short textual identifier of the feature unit
name	The name of the feature
context	The administrative aspects of the feature object
releasabilityType	This attribute is optionally used to differentiate between releasability status for particular features
zoneObjectType	This attribute identifies which Feature Unit Location Object type this object refers to based on the Feature Catalogue: <ul style="list-style-type: none"> - Internal Water - Archipelagic Waters - Territorial Sea - Contiguous Zone - Exclusive Economic Zone - Continental Shelf - Roadsteads - High Seas - The Area
areaValue	This attribute describes the value of the area of the zone

Attribute Name	Description
jurisdictionDomainType	Defines the vertical juridical domain of the object: - Airspace - Land Surface - Water Surface - Water Column - Seabed Surface - Subsoil
areaType	This attribute describes the type of the area of the zone: - Official Area - Nonofficial Area - Calculated Area - Surveyed Area
areaSurfaceRelation	Relationship to surface: - Mixed - Below - Above - On Surface
areaReferencePoint	The coordinates of a point inside the spatial unit
surfaceRelation	Relationship to surface
areaUnitOfMeasure	The unit of measure used to define the area of the "Zone"
administrativeUnit	Reference to the Basic Administrative Unit to which this feature is associated
source	Reference to the source object
surface	Reference to the associated surface geometric object
boundary	Reference to limit or limits that define the outer boundary of zone
innerBoundary	Reference to limit or limits that define one or more inner boundary of zone (holes such as for islands)

3.5.6.4 Point Block

The Point block is an object that defines the spatial geometry of a Point.

Table 3-15 – PointBlock - optional

Attribute Name	Description
sald	The spatial attribute identifier in the Oid format
locationByText	This attribute allows a spatial attribute to be a textual description. This allows "Location", "Limit", "Zone"s and "Space" that are not fully described geometrically to be included Quantity Specifications
referenceSystem	Allows a CoordinateReferencingSystem (CRS) to optionally be specified. In many other S100 based products the CRS is only defined at the metadata level and applies for the whole data set; however, in S-121 it is necessary to detail it right down to the specific instances of geometry since treaty points and lines may come from different sources such as different treaties that may use different CRS
originalLocationTextualPosition	Reference location of a point when it was first established
originalLocationReferenceSystem	Reference System for a Reference location of a point when it was first established
officialLocationTextualPosition	Official location reference of the point
officialLocationReferenceSystem	Reference System for an official reference of the point
globalLocationTextualPosition	Global reference to a point in a world global datum such WGS84 or ITRF2000
globalLocationReferenceSystem	Reference System for a global reference to a point in a world global datum such WGS84 or ITRF2000

Attribute Name	Description
WGS84LocationTextualPosition	WGS84LocationTextualPosition
commonLocationTextualPosition	Common reference location of point in a datum that is defaulted to the other points in the dataset
commonLocationReferenceSystem	Reference System for a common reference location of point in a datum that is defaulted to the other points in the dataset
pointGeometry	GM_Point geometry as per IHO S-100 Part 7 Spatial Schema. The associated data type is structured as a formatted Character String to assist in implementation; however, a GIS system encode a Direct Position as an internal numeric type and the type can be converted to a character string upon encoding
scaleMaximum	Maximum Scale at which the object should be portrayed
scaleMinimum	Minimum scale at which the object should be portrayed
source	Reference to the source object

3.5.6.5 Curve Block

The Curve block is an object that defines the spatial geometry of a Curve.

Table 3-16 – CurveBlock - optional

Attribute Name	Description
salD	The spatial attribute identifier in the Oid format
locationByText	This attribute allows a spatial attribute to be a textual description. This allows "Location", "Limit", "Zone" and "Space" that are not fully described geometrically to be included Quantity Specifications
referenceSystem	Allows a CoordinateReferencingSystem (CRS) to optionally be specified. In many other S-100 based products the CRS is only defined at the metadata level and applies for the whole data set; however, in S-121 it is necessary to detail it right down to the specific instances of geometry since treaty points and lines may come from different sources such as different treaties that may use different CRS
curveGeometry	Polyline or arcByCentrePoint or circleByCentrePoint formatted character string consisting of: <ul style="list-style-type: none"> - Sequence of vertices (longitude coordinate, latitude coordinate or Direct Position), or - arcByCentre (arcStart longitude, slash delimiter, arcStart latitude, comma delimiter, arcEnd longitude, slash delimiter, arcEmd latitude, comma delimiter, arcCentre longitude, slash delimiter, arcCentre latitude), or - circleByCentre (circleCentre longitude, slash delimiter, circleCentre latitude, comma delimiter, circlePoint longitude, slash delimiter, circlePoint latitude)
wGS84CurveGeometry	Polyline or arcByCentrePoint or circleByCentrePoint formatted character string consisting of: <ul style="list-style-type: none"> - Sequence of vertices (longitude coordinate, latitude coordinate or Direct Position), or - arcByCentre (arcStart longitude, slash delimiter, arcStart latitude, comma delimiter, arcEnd longitude, slash delimiter, arcEmd latitude, comma delimiter, arcCentre longitude, slash delimiter, arcCentre latitude), or - circleByCentre (circleCentre longitude, slash delimiter, circleCentre latitude, comma delimiter, circlePoint longitude, slash delimiter, circlePoint latitude)
scaleMaximum	Maximum Scale at which the object should be portrayed
scaleMinimum	Minimum scale at which the object should be portrayed
source	Reference to the source object

3.5.6.6 Surface Block

The Surface block is an object that defines the spatial geometry of a Surface.

Table 3-17 – SurfaceBlock - optional

Attribute Name	Description
salD	The spatial attribute identifier in the Oid format
locationByText	This attribute allows a spatial attribute to be a textual description. This allows "Location", "Limit", "Zone"s and "Space" that are not fully described geometrically to be included Quantity Specifications
referenceSystem	Allows a CoordinateReferencingSystem (CRS) to optionally be specified. In many other S-100 based products the CRS is only defined at the metadata level and applies for the whole data set; however, in S-121 it is necessary to detail it right down to the specific instances of geometry since treaty points and lines may come from different sources such as different treaties that may use different CRS
scaleMaximum	Maximum Scale at which the object should be portrayed
scaleMinimum	Minimum scale at which the object should be portrayed
source	Reference to the source object

Appendix A Implementation Schema

(Normative)

The implementation schema is a derived schema from the schema presented in the S-121 Product Specification document. The S-121 schema is a hierarchical model that builds on the S-100 model and the ISO 19152 Land Administration Domain Model. The S-121 schema presented in the Product Specification document reflects this structure of classes being derived from other classes. There is a significant depth to the hierarchy of classes within classes due to the multiple sources.

In order to make implementation easier an implementation schema has been developed. This schema deliberately flattens the structure pulling many linked and shared objects into classes as simple attributes. Also many data types have been simplified to allow them to be included in character strings to allow them to be output in an Explicit Text Format. The following figures present all of the classes from the S-121 model together, on the right with equivalent implementation classes.

An example of the simplification of classes is the collapsing of the Object Identifier class Oid from a class with two parameters to a structured single character string of a particular format. Figure A-1 shows the implementation of the class Oid as Oid_FormattedCharacterString.

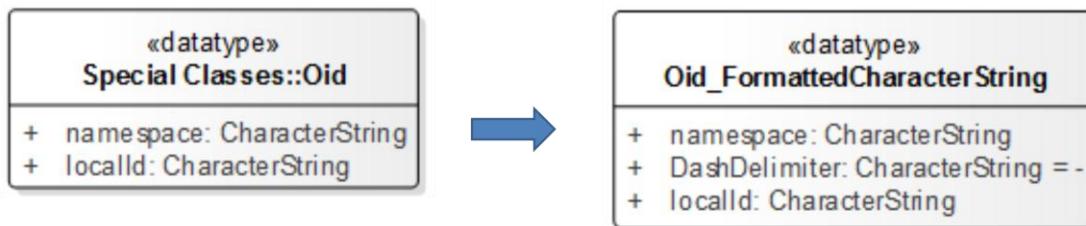


Figure A-1 – Mapping of Oid to an Implementation Class

In an ArcGIS or other GIS system specific implementation the process may be refined one more step. The implementation classes may further be mapped to classes in the GIS system specific structure. This further level of mapping is not shown here because it is GIS system dependent.

The S-100 standard allows for the specialization of features and information objects by the addition of attributes and the definition of any number of thematic attributes; however, it does not allow for the specialization of spatial attributes. This restriction is required to support display systems such as the Electronic Chart Display Information System (ECDIS). S-121 includes additional ways of describing position inherited from the ISO standard 19152. These are the description of a spatial attribute textually called "location by text" and the description of a position in a coordinate reference system different than that used to describe other positions. Both cases occur in real data. In order to ensure that the spatial information complies with S-100 an additional attribute class has been defined to carry the additional spatial attributes.

The following series of figures are taken from the S-121 UML model and show the implementation classes related to each of the classes in the high level S-121 schema.

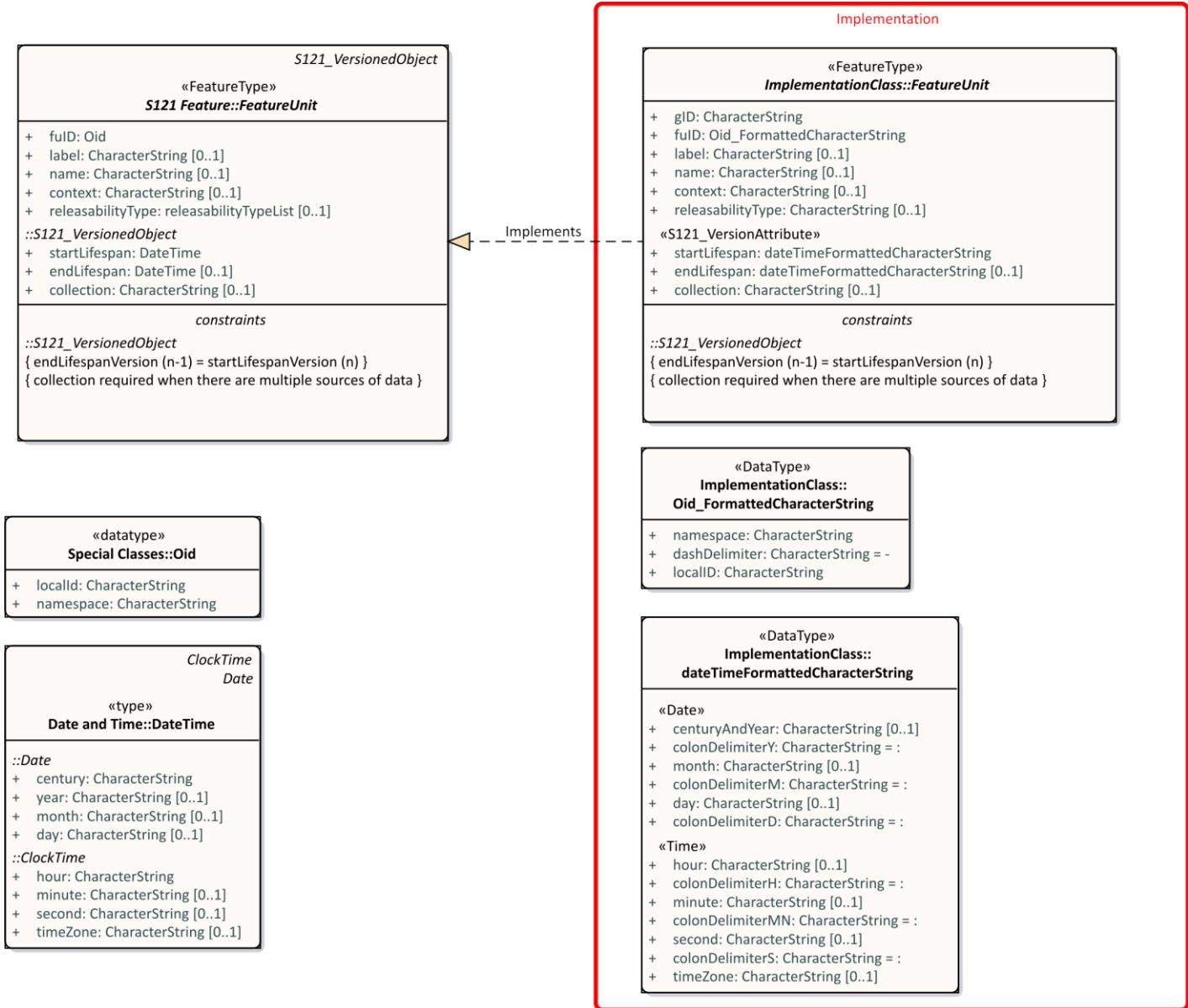
**Figure A-2 – Implementation Class - S121_FeatureUnit**

Figure A-3 shows the Spatial Attribute Implementation classes.

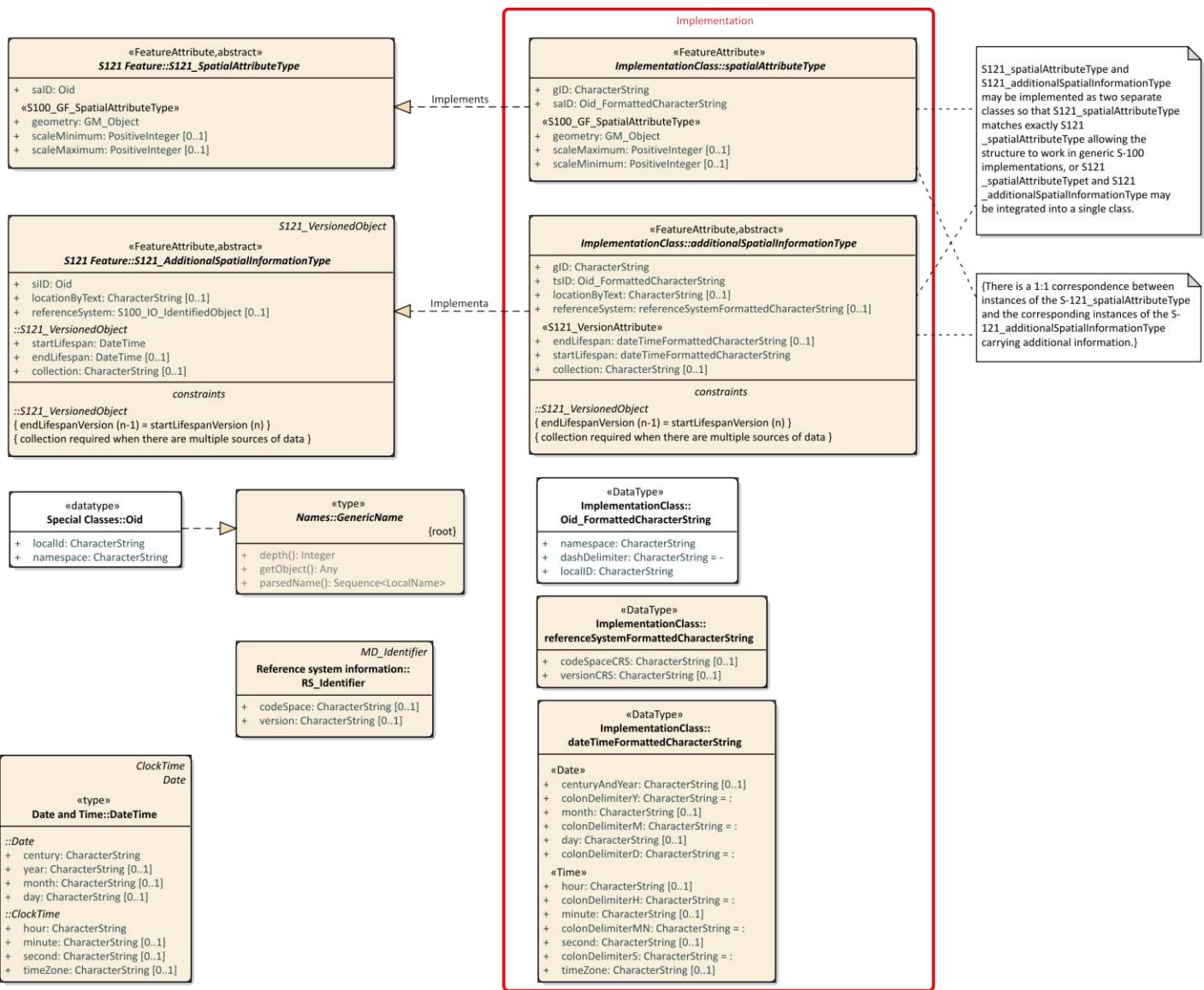


Figure A-3 – Implementation Class - S121_SpatialAttribute

Figure A-4 shows the Location class. The geometry has reduced to as single Direct Position which is shown as a formatted character string.

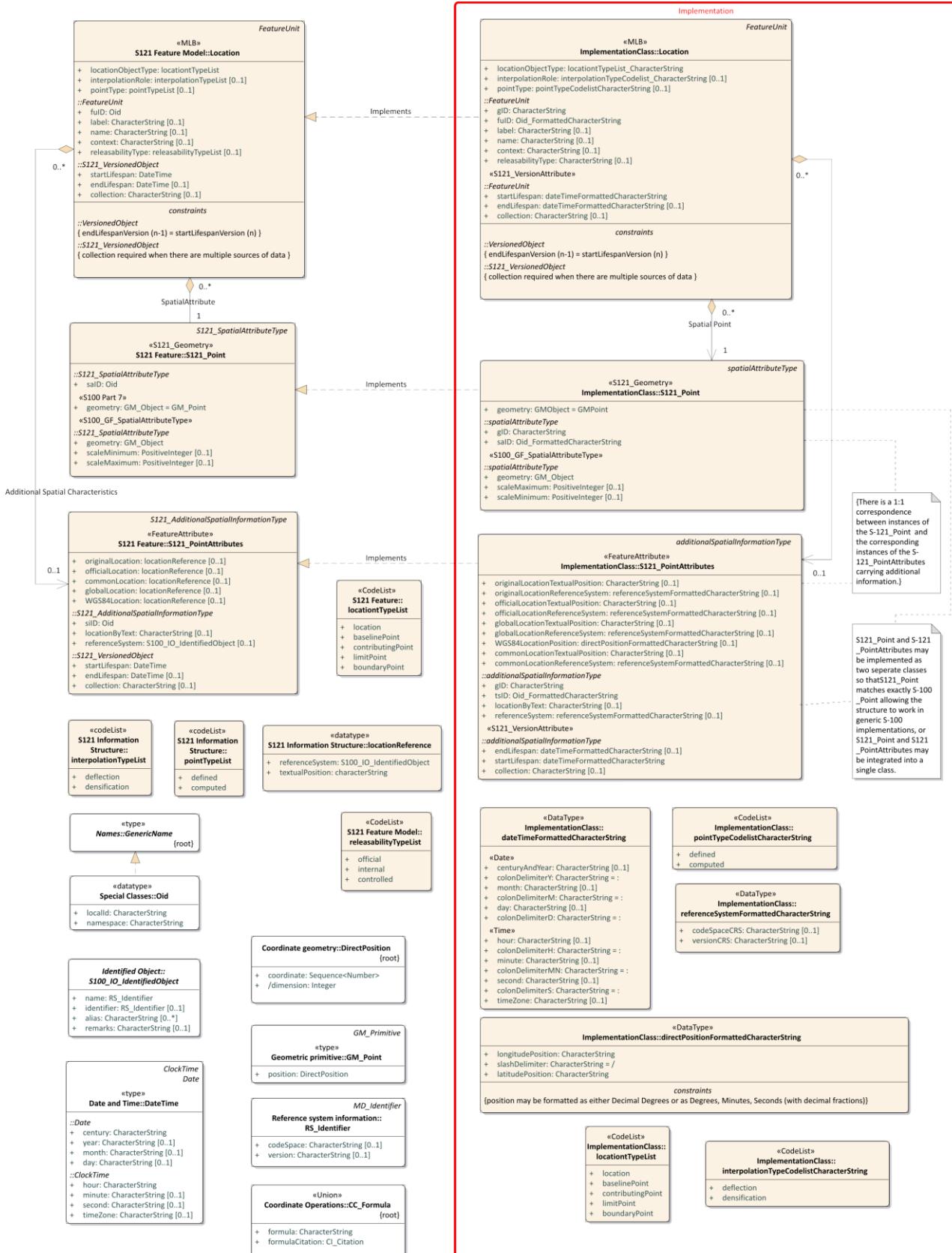


Figure A-4 – Implementation Class - S121_Location

Figures A-5 shows the Limit class. The geometry has been reduced to a structured PolyLine or Arc. A vertex string is a structured character string. The Arc by Centre and Circle by Centre structures are also allowed since they are supported in S-100.

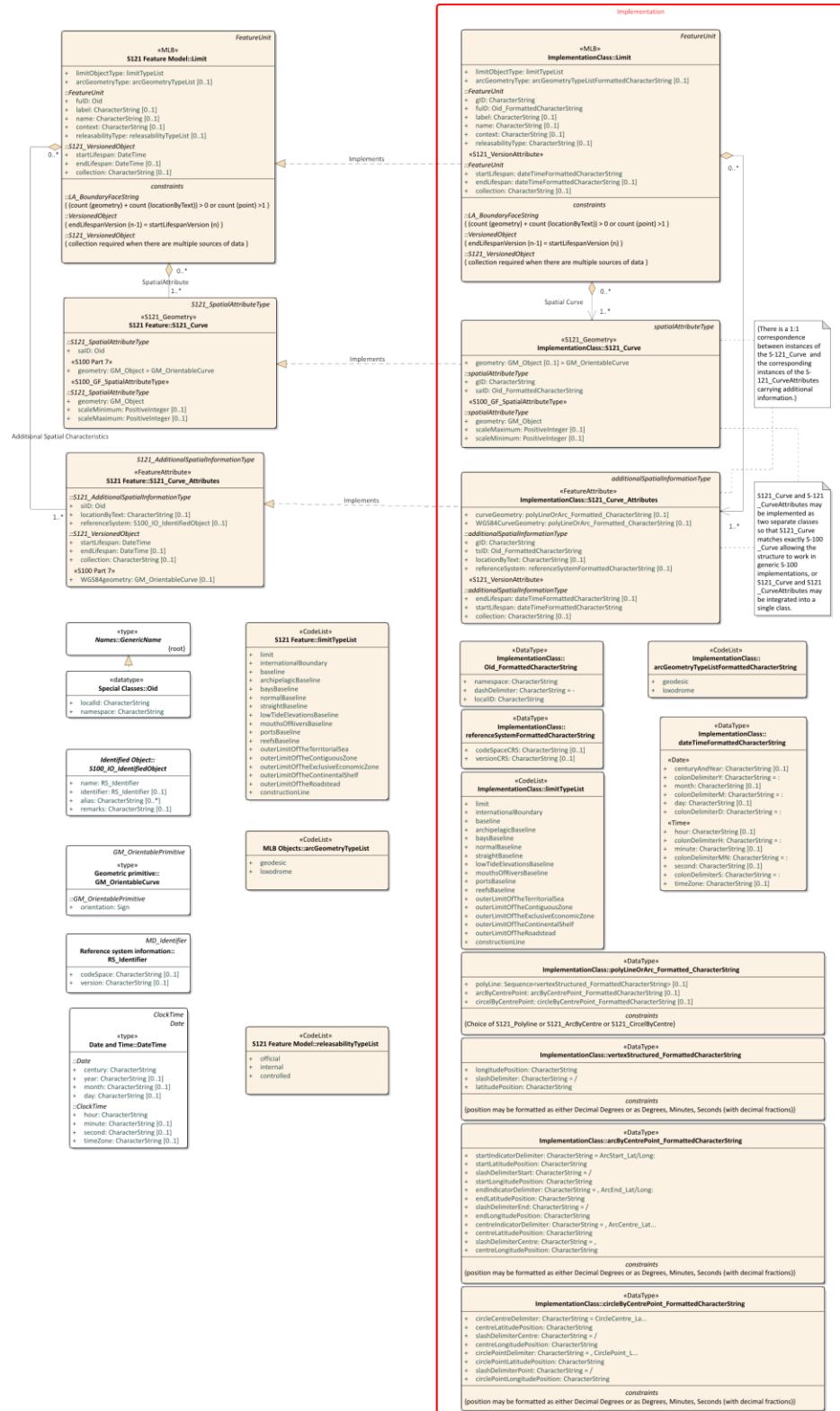


Figure A-5 – Implementation Class - S121_Limit

Figures A-6 shows the Zone class. The geometry is done by Zone to Limit relationships. The S121_Surface object carries attributes about the geometry but not the actual geometry.

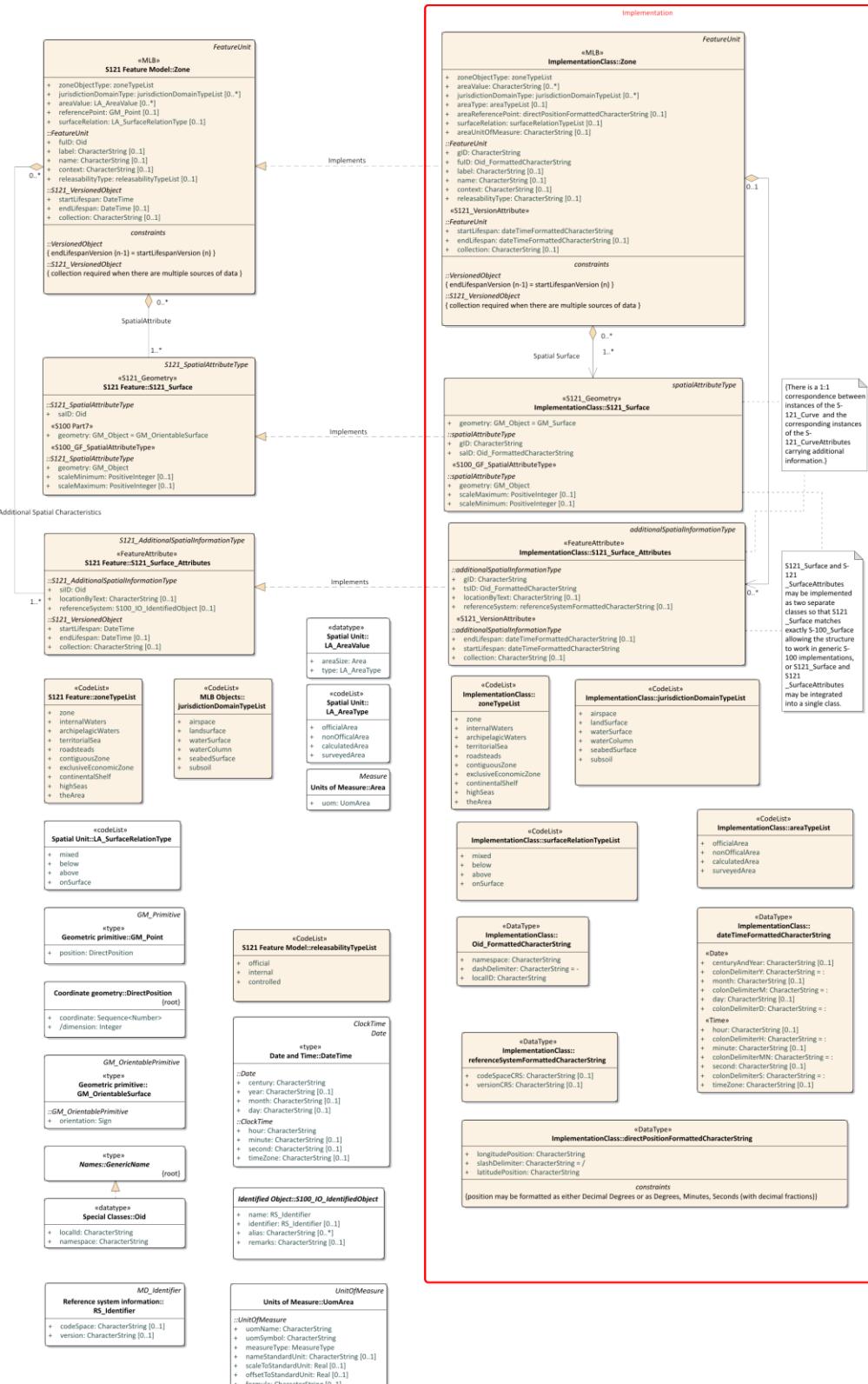


Figure A-6 – Implementation Class - S121_Zone

Figures A-7 shows the Space class. Space is essentially a zone with a reference point and a height. The position is a direct position and a vertical.

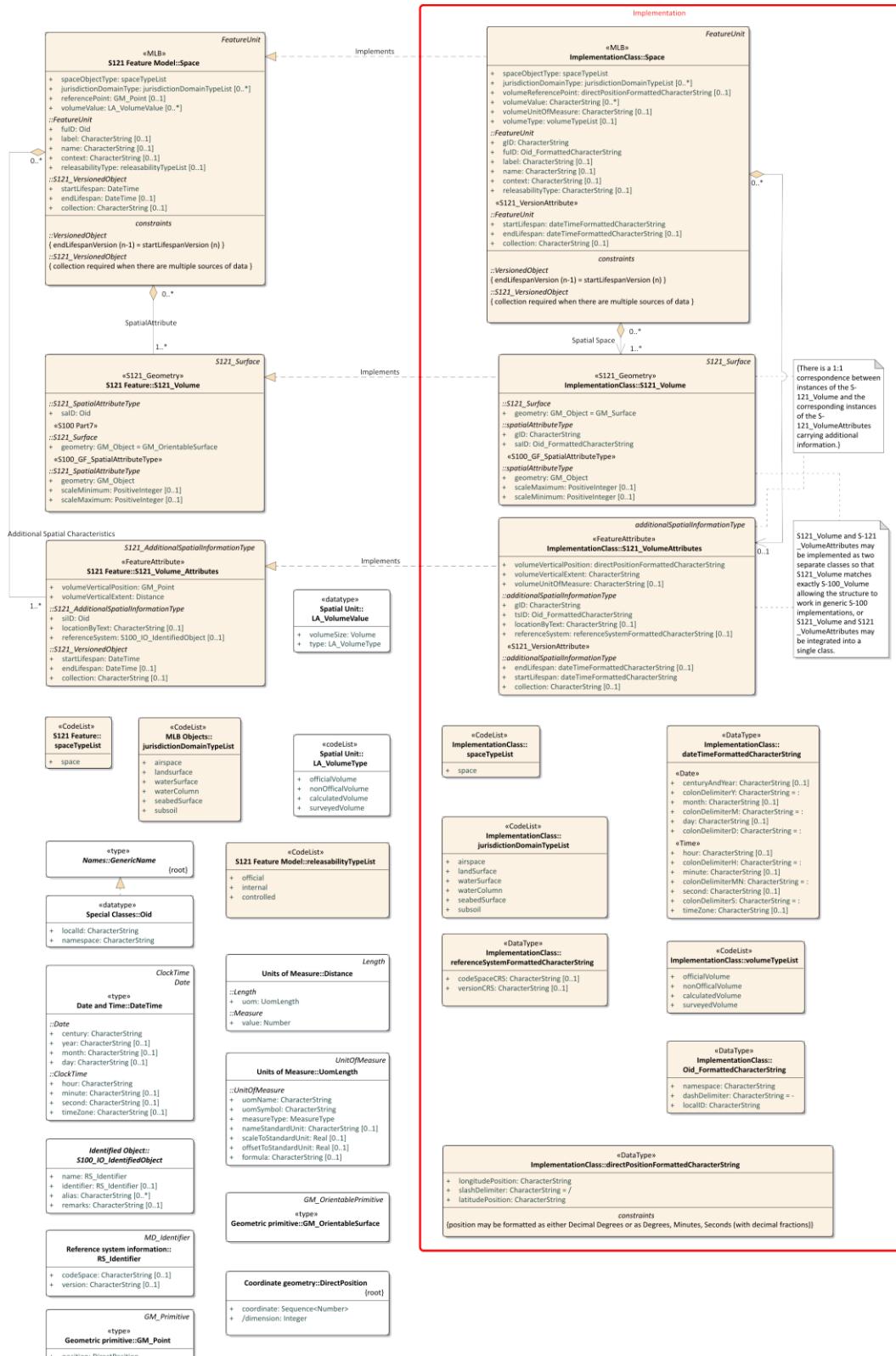


Figure A-7 – Implementation Class - S121_Space

Figure A-8 shows the implementation of the Basic Administrative Unit class. This is a simple class with only a few character string attributes.

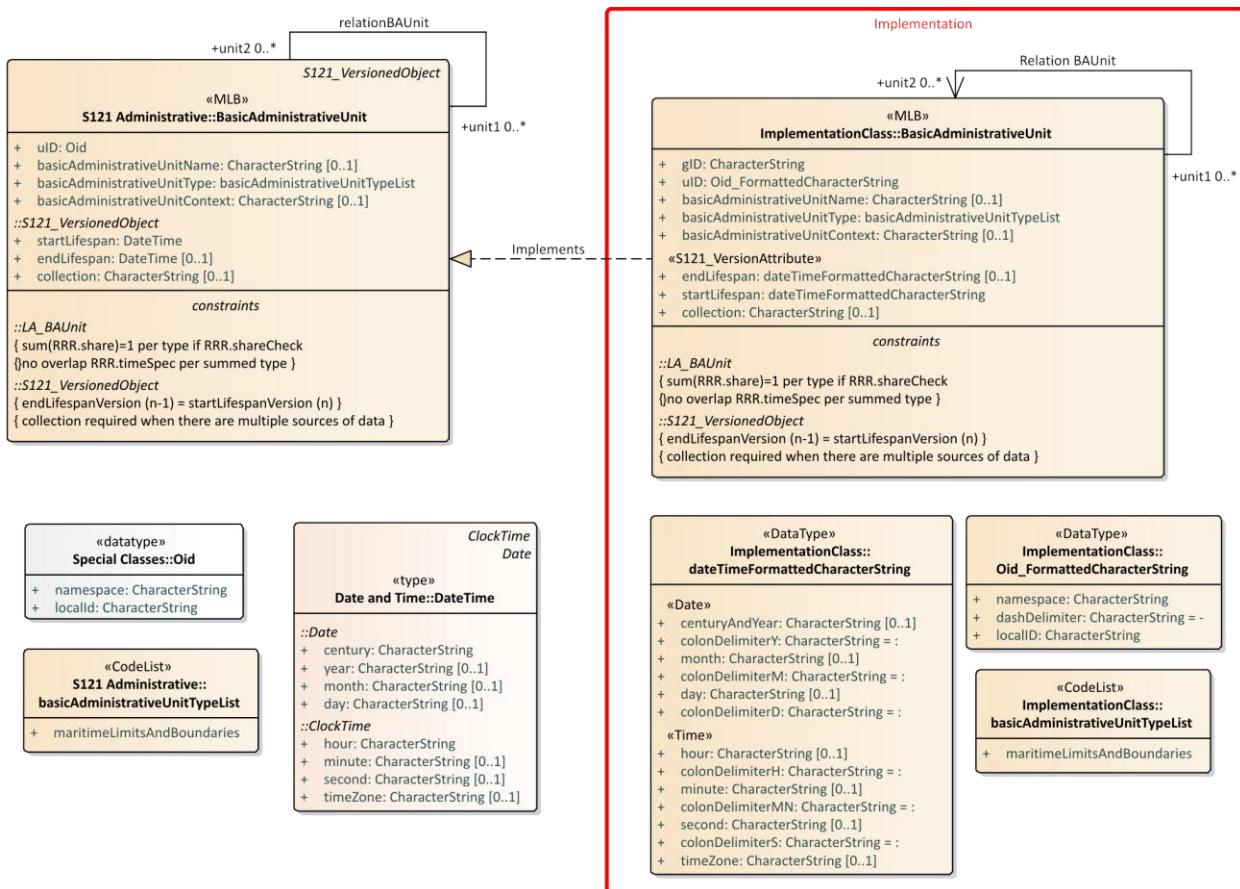


Figure A-8 – Implementation Class - S121_BAUnit

Figure A-9 shows the implementation of the Governance class. This simple class remains essentially unchanged in the implementation model.

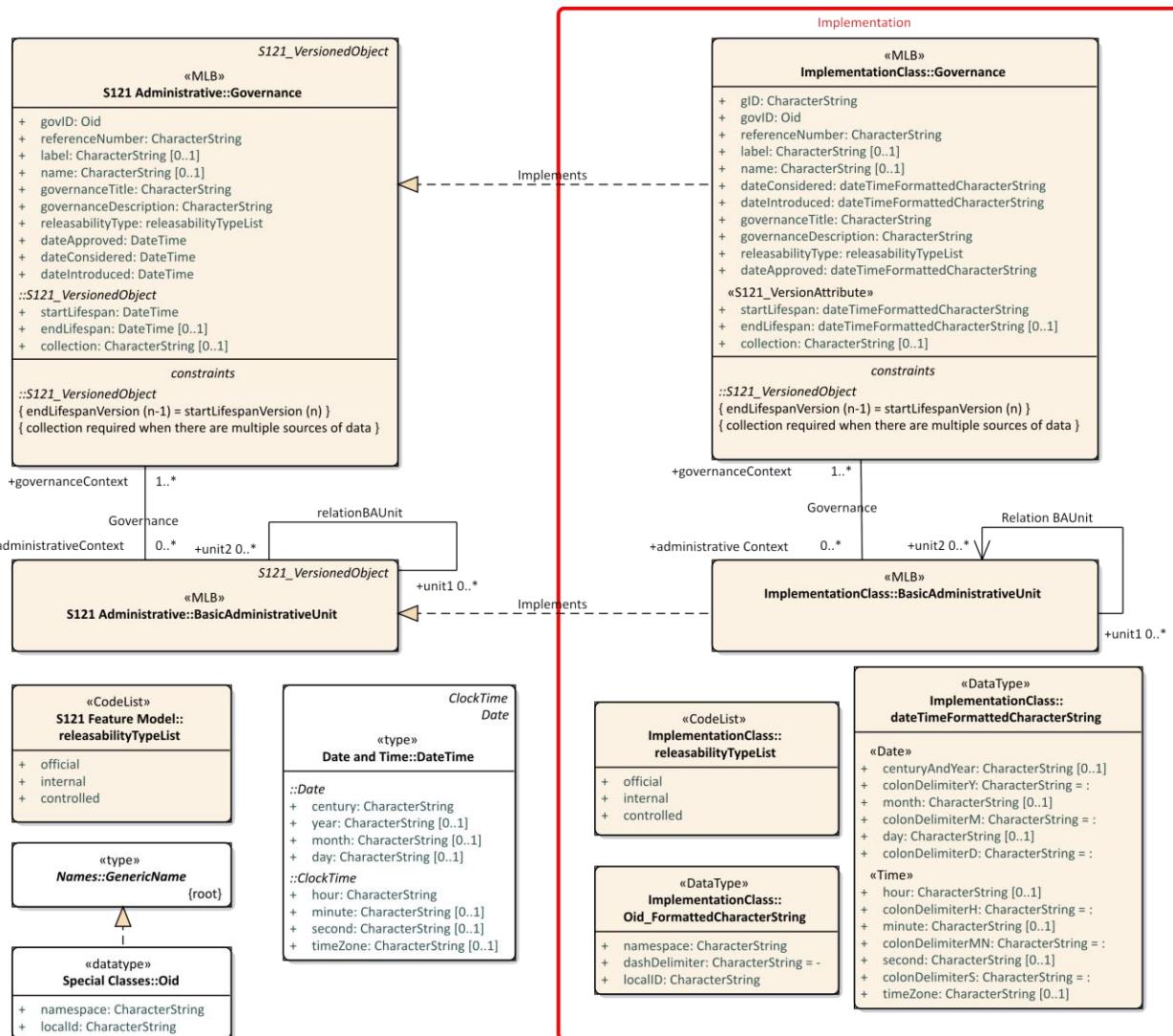


Figure A-9 – Implementation Class - S121_Governance

Each of the Rights, Restrictions and Responsibilities classes are treated separately. This is shown in Figures A-10, A-11 and A-12. The MA_RRRshare attribute is shown as a Fraction data type and MA_RRRshareCheck is shown as Boolean. The fraction and Boolean also need to be represented as character string elements.

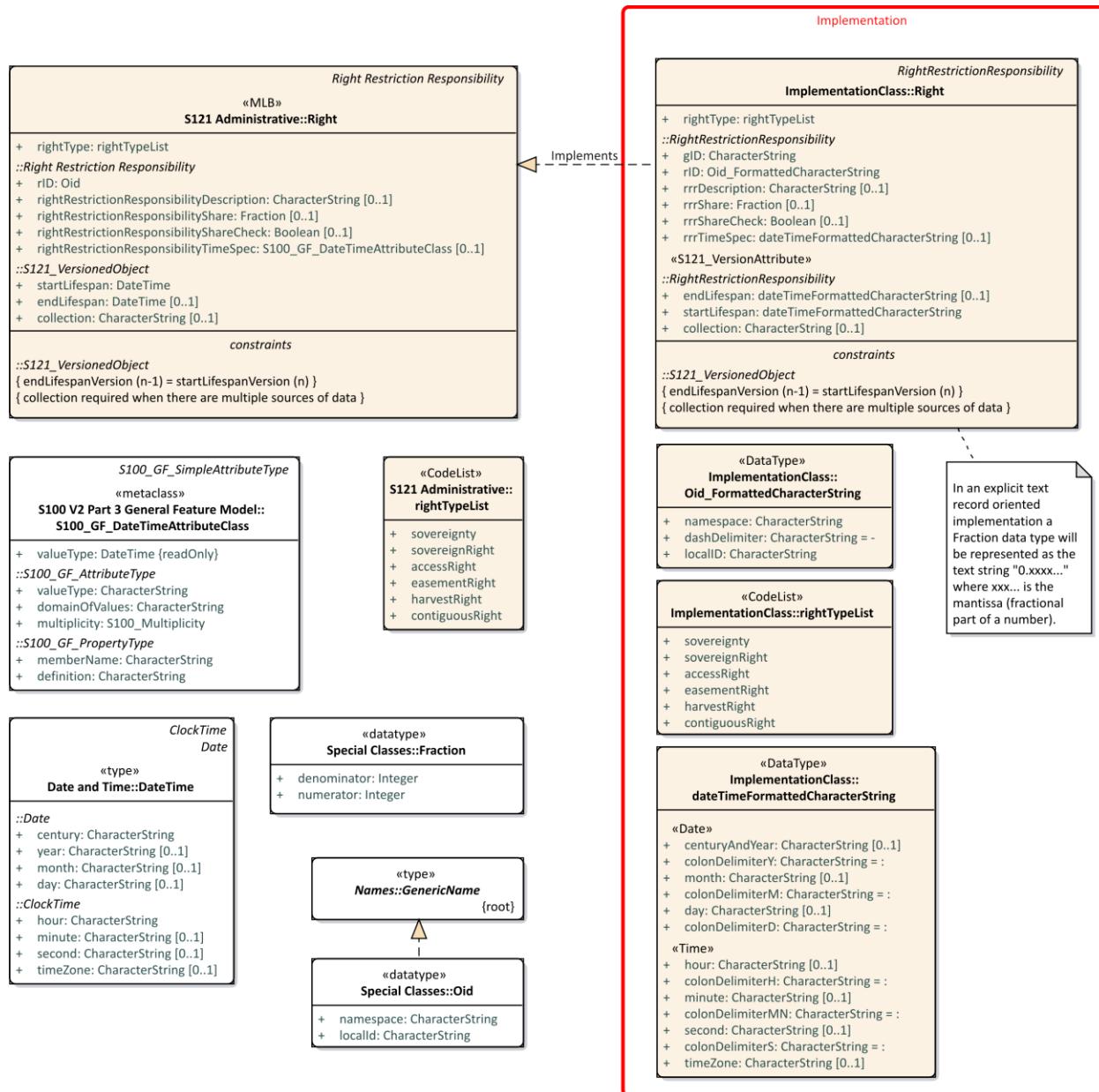


Figure A-10 – Implementation Class - S121_Right

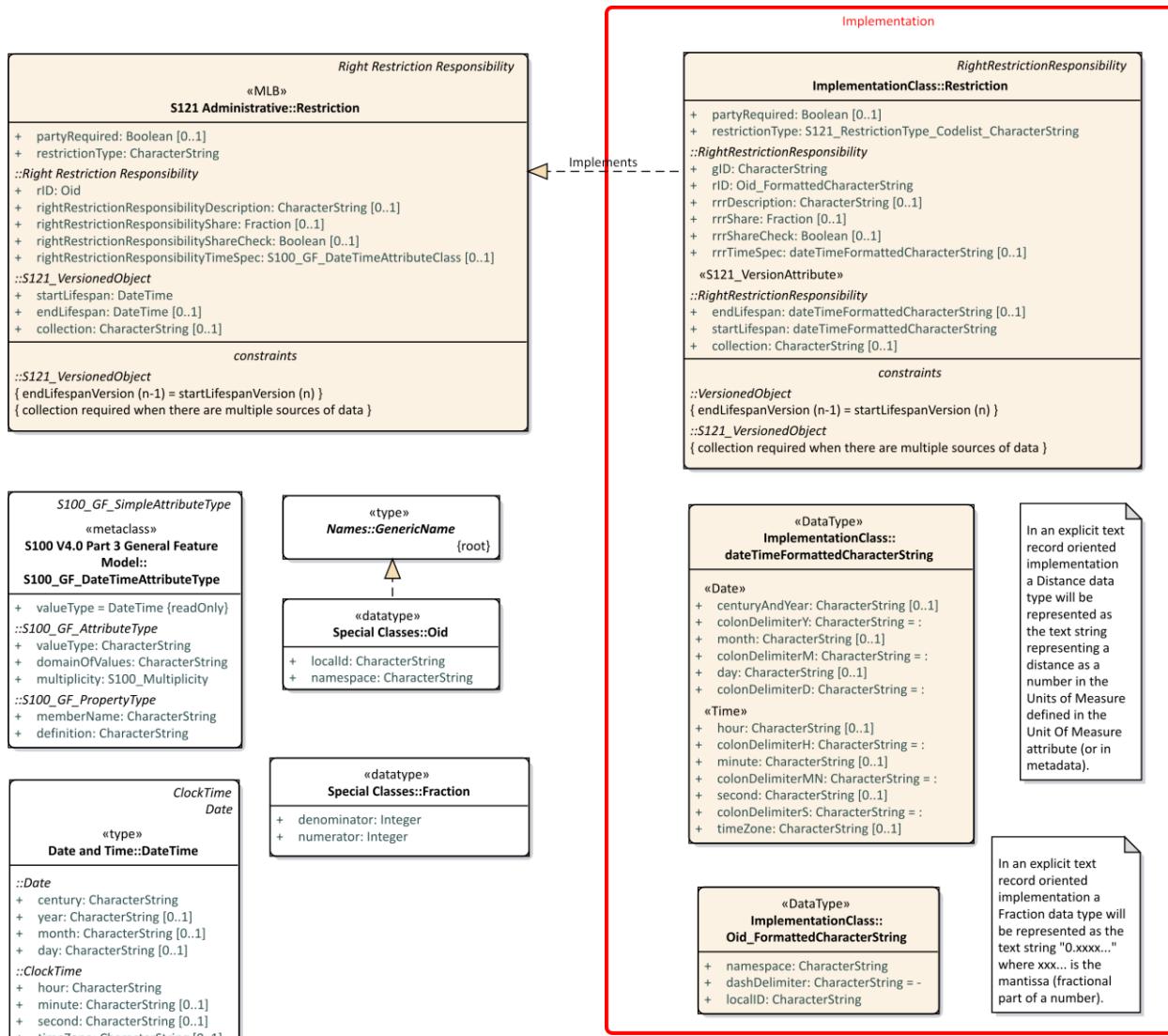


Figure A-11 – Implementation Class - S121 Restriction

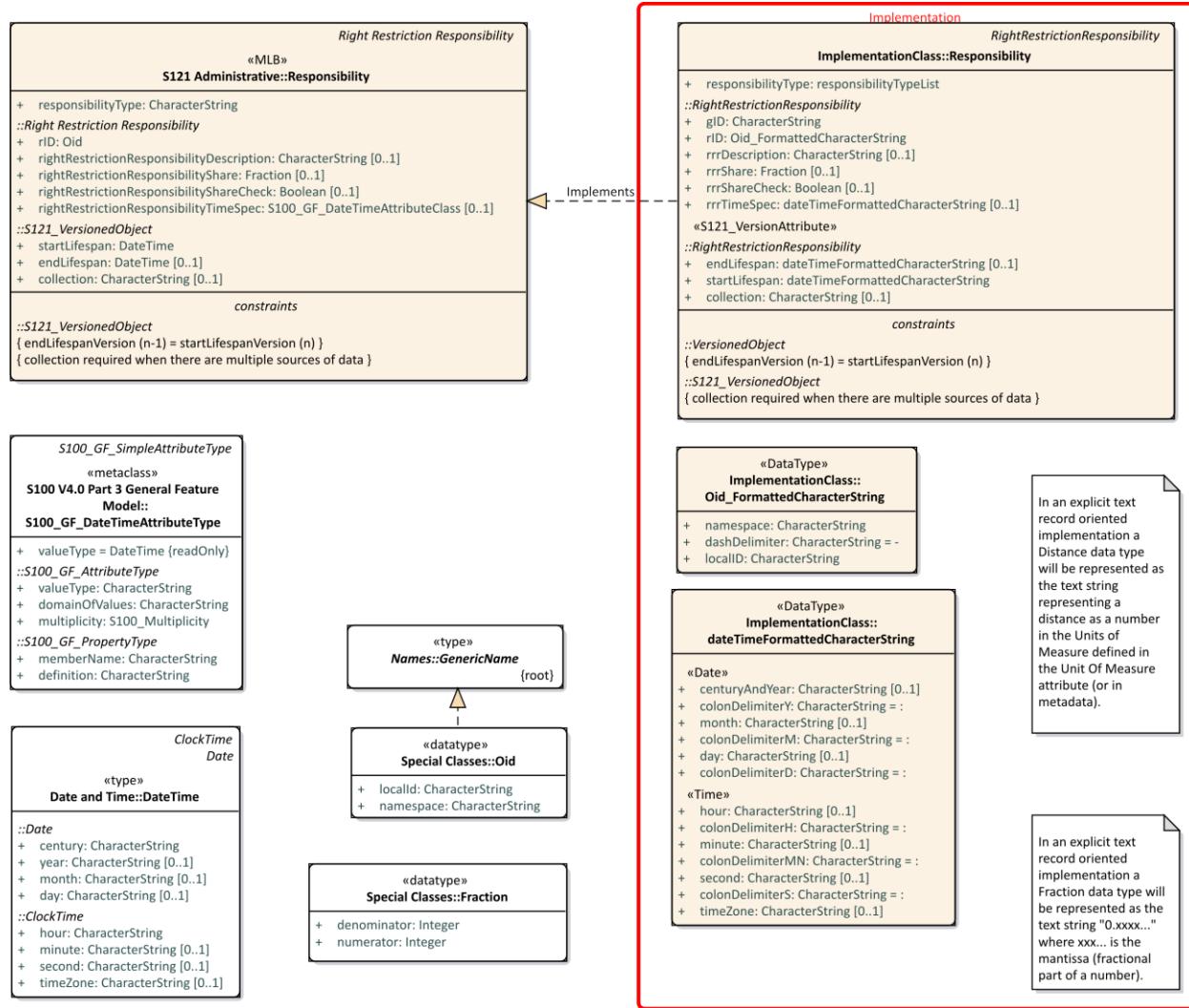


Figure A-12 – Implementation Class - S121_ Responsibility

Figure A-13 shows the S-121_PartyGroup classes. The composition relationship between S121_Party and S121_GroupParty can be implemented as a pointer (or set of pointers). The optional S121_PartyMember MA_partyShare attribute can be implemented as one to three attributes following the pointer.

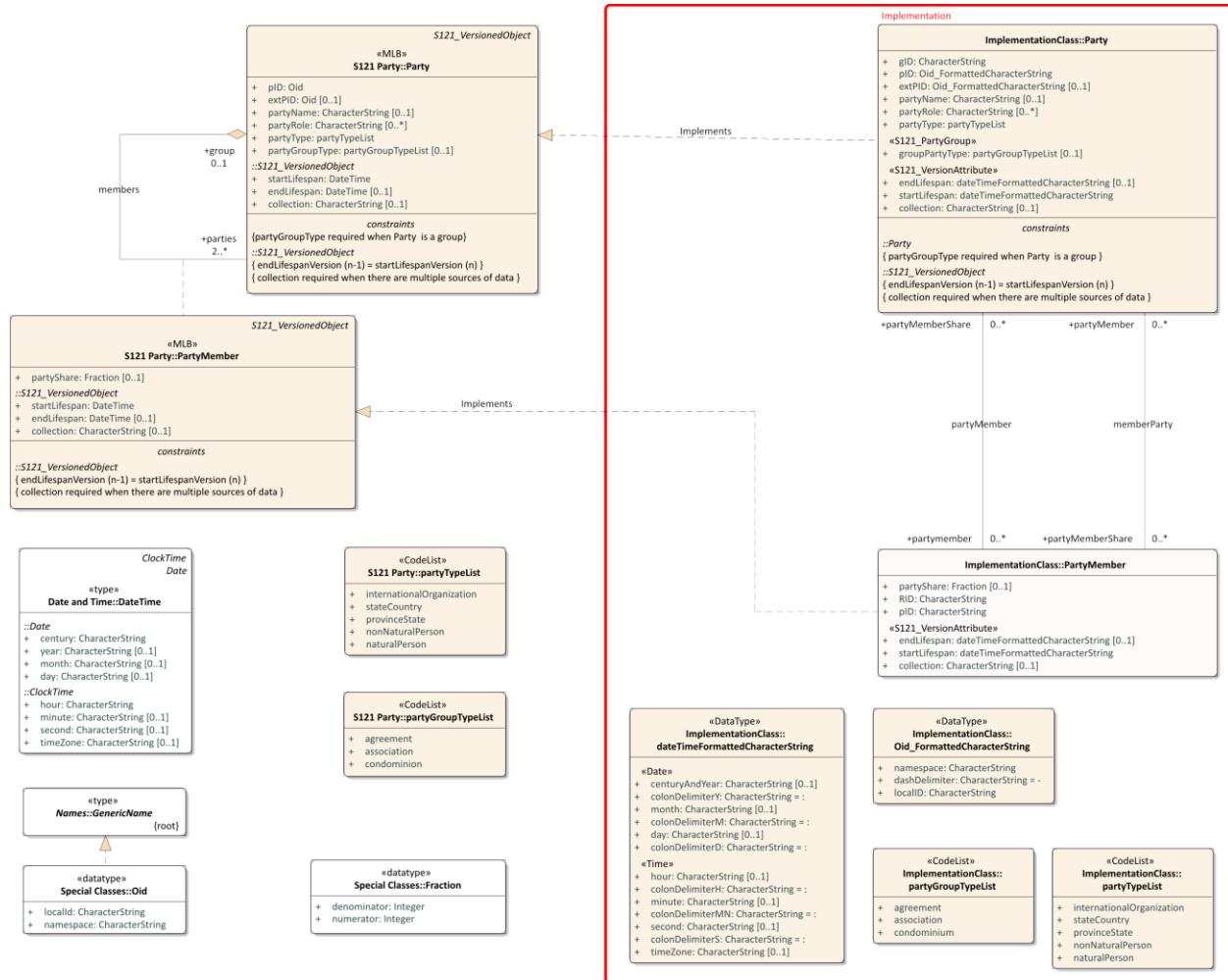


Figure A-13 – Implementation Class - S121_PartyGroup

Figure A-14 shows the implementation of the class Source. This complex structure has been flattened to a single class with a lot of attributes, most of which are optional.

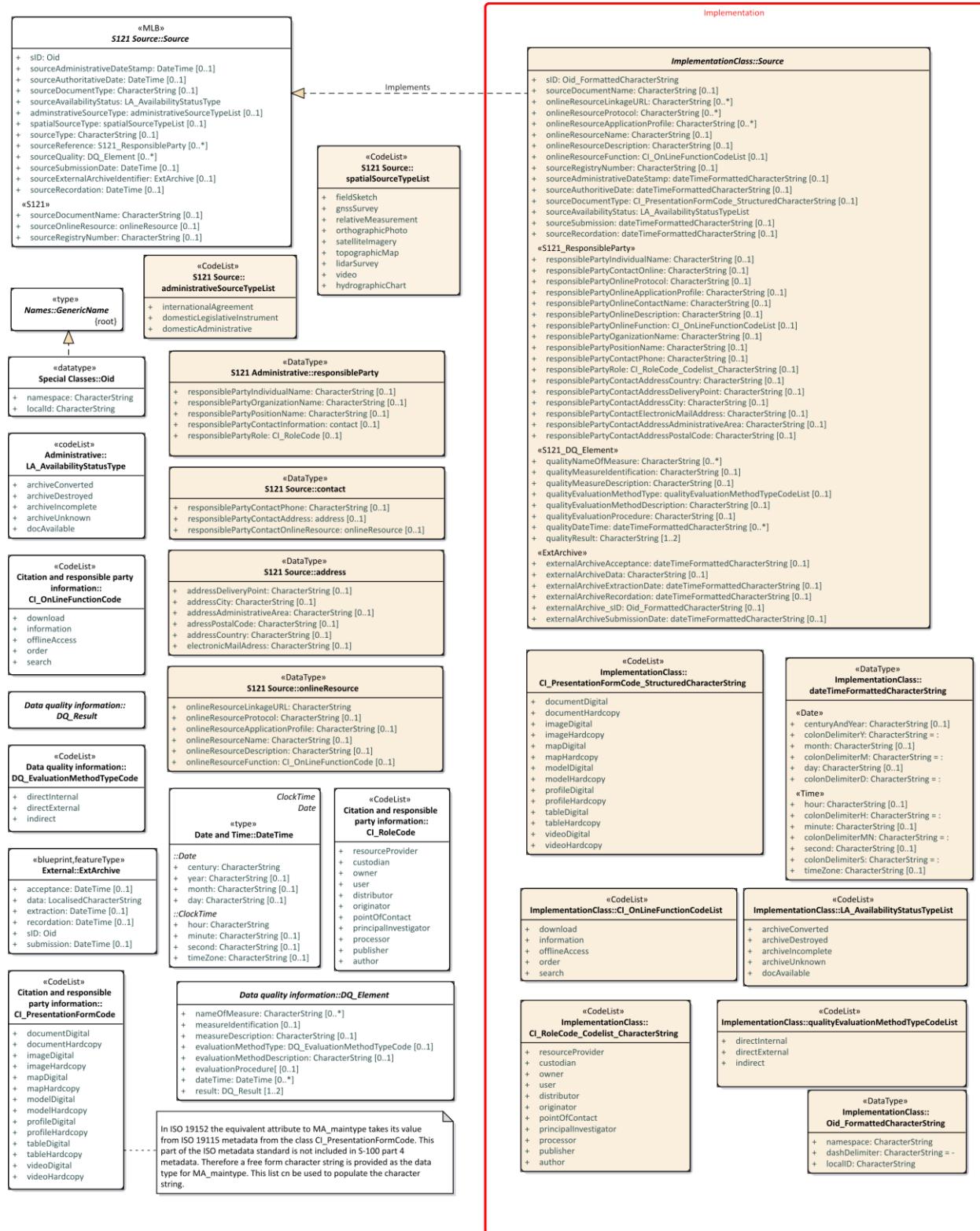


Figure A-14—Implementation Class - S121_Source

Appendix B The Carriage Return / Line Feed Delimiter

(Informative)

Record oriented communications systems have used the CR LF combination – Carriage Return CR (IA5 Code 13) followed by Line Feed LF (IA5 Code 10) since the earliest days of computing. The combination derives from the Model 33 Teletype machines that were used as the terminals on early computers. Since these terminals were physical printers there was need for time for the print carriage to return to the beginning of the line before it was possible to print the next line. The Teletype machine took up to 200 milliseconds to move the carriage back to the beginning of the line. At the print rate of 10 characters per second the combination CR followed by LF returned the carriage and moved to the next line within the time window.

All of the early computer systems adopted CR LF as the delimiter for a character string. Later when computer output was on a video display some of the computer systems changed for efficiency. There is only a need for a single character to delimit the end of a character string or line. In those days storage space was precious.

Unfortunately all computer companies have not been consistent. Microsoft uses the CR LF combination – Carriage Return CR (IA5 Code 13) followed by Line Feed LF (IA5 Code 10). Apple used CR alone, and Unix (and recent Apple systems) use LF alone. Most text processors accept either CR or LF or the combination CR LF.

This confusion remains to this day where some data is delimited with CR and other with LF. The CR LF combination seems to be read and interpreted by all systems. Therefore S-121 should use CR LF as a character string delimiter; however, one should expect that some display systems may print two blank lines between character strings rather than one. This does not affect the readability.

Page intentionally left blank

Appendix C. Abstract Syntax Notation

(Informative)

C 1. Introduction

The ISO 8824 Abstract Syntax Notation (ASN.1) is a meta language used to describe a data interchange format in a context-independent manner. ASN.1 itself is similar to the compiler description languages used in computer science to define programming languages. In fact the structure is similar to a programming language and a particular data file is analogous to a particular program written in the language. Like a language, the definition of a data file makes use of only those words in the language which apply in a given situation. Words, or statements composed of sequences of words, may be used as often as needed, in accordance with the syntactic rules of ASN.1 and the information model being encoded.

A number of basic data types make up the "vocabulary" of the "language". A complete description of the Abstract Syntax Notation would take some space to present. However in this document only a subset of the ASN commands need be used. This subset is identified below:

- Assignment
- CHOICE
- SEQUENCE
- SEQUENCE OF
- SET
- SET OF
- MACRO

Quite complex data types can be described. This is accomplished by defining a small number of simple data types. The complete range of values of these simple data types may be defined, and then the manner of combining these simple data types into the more complex data types are specified. A data type defined in this manner may be assigned a tag so that it may be identified in the communication or otherwise distinguished. These tagged elements correspond to the classes in the UML model. Encoding rules described separately from the syntax of the data format define the manner in which the data types are encoded and delimited. There are several sets of encoding rules established by ISO for direct use with ASN.1. A set of binary encoding rules are described in ISO 8825; however, the syntax notation is not locked to any one particular set of encoding rules and any of the encoding rules described in the annexes of this document may be used to represent data corresponding to the syntax described.

C 2. Syntactic Structure

The Abstract Syntax Notation defines a communications format in terms of a syntactic tree. At the highest level the entire data set which is interchanged is described as a sequence of sub-sections. These sub-sections are then broken down into their component elements and so on, until each of the primitive data elements is identified. Each data element has a particular data type such as a basic text string, an integer etc. When data is communicated it is in terms of the syntactic tree. Many branches of the syntactic tree may be optional. When encoding using a tag oriented encoding technique such as ISO 8825 or XML the data stream is parsed by the receiving device and matched to the elements of the syntactic tree. In a tag oriented exchange formats, it is not necessary to pad the communications with null fields to accommodate information fields which are not used. The tag numbers associated with data elements allow the parser to skip over sections of the syntactic tree, so that only the relevant information need be communicated. This approach also allows for recursive definitions, so that there is no limit to the number or length of parameter data.

The S-121 Explicit Text Format is a tag oriented encoding. The tags are the Record Type Identifiers at the beginning of each record. Each record is terminated. In the S-121 Explicit Text Format each record is separate so there is no need for recursive structures.

C 2.1. Assignment

Branches of the syntactic tree are defined in terms of expressions in the Abstract Syntax Notation. The Assignment operator ::= equates a reference name to a series of more basic commands or primitive data elements. An entire syntax is described in terms of successive refinement. A data file consists of a number of named sections, each of which consists of a number of sub-sections etc. down to the level of the primitive data elements.

C 2.2. SEQUENCE and CHOICE

The basic operator in an assignment statement is the **SEQUENCE** command. The SEQUENCE operator specifies that a branch of the syntactic tree consists of a defined number of elements in a fixed order; that is, a list of mandatory or optional elements. The **SEQUENCE OF** operator specifies that the number of elements is variable but that the order is fixed; that is, it indicates a repeated element or list of elements. The **SET** operator specifies that a branch of the syntactic tree consists of a defined number of elements in any order. It is equivalent to a sequence but the order is not fixed. The **SET OF** operator specifies that the number of elements is variable and that the order is variable. The **CHOICE** operator specifies that an instance within the syntactic tree permits the inclusion of one data element (or sub-branch) out of a fixed set of choices.

C 2.3. OPTIONAL and DEFAULT

Elements in a SEQUENCE or SET may be optional. This is, indicated by the keyword **OPTIONAL** immediately following the element name in the sequence or set. If a default value is defined for an optional element then the keyword **DEFAULT** replaces the term OPTIONAL and the value of the default is specified following the DEFAULT keyword.

C 2.4. MACRO

A **MACRO** definition allows a primitive data type to be constructed out of other primitive data types. It also allows a default value notation to be defined for the constructed primitive so that the DEFAULT specification may be used with this new element.

C 2.5. Tag Numbers

Tag numbers are used to identify branches of the syntactic tree. A tag number is expressed in square brackets [] preceding an element name. Tag numbers are not necessary in a fixed SEQUENCE with no OPTIONAL elements since the order is known; however, in other cases, tag numbers must be supplied for each element. There are four classes of identifier tag codes: Universal, Context Specific, Application, and Private. **Universal tags** are used to identify operators such as CHOICE or SEQUENCE or basic data elements such as a GRAPHIC STRING. **Context Specific tags** are assigned in each branch of the syntactic tree to number the elements of the tree in that branch. The same tag numbers are used over and over in different levels of the syntactic tree, and it is the responsibility of the syntax parser interpreting data encoded in terms of the ASN defined syntax to keep track of the levels of the tree and the local meaning of each tag. **Application wide tags** are used to define unique identifiers to specific elements. A parser does not need to keep track of the level of the syntax to identify the particular data element. Since Application wide tags are unique to a particular data element, long tag numbers would result if they were used extensively throughout a particular syntax. Therefore their use should be restricted to a few strategic places within the syntax. **Private tags** are used to build proprietary extensions to ASN.1, and are rarely used.

In certain situations redundant tags are generated. For example, a Universal tag may immediately follow a Context Specific tag. The keyword **IMPLICIT** is used to suppress the generation of a universal tag in those situations where the tag is unnecessary. An **IMPLICIT SEQUENCE** of two tagged elements would be encoded using only the context specific tags identifying each of the elements. The tag identifying the overall sequence would not be included.

C 2.6. Grammar

The following is a summary of the notations used in an example grammar.

::= - is the production symbol of the grammar and can be read "is produced by" or "is composed of". A production allows the definition of a syntactic entity by the assignment of a name to a collection of data types or other entities. Entities placed on the right of the production symbol are taken together to define the composite entity. By the use of production statements the elements of the grammar may be defined by successive refinement.

CHOICE { } - is the alternative indicator in a production definition. It can be read as "or". Alternate data entities are separated by commas within the brackets of the CHOICE indicator. Each of these entities must be identified by a TAG so that the particular entity communicated may be distinguished. For example, the data entity Type consists of a choice of data element Type-1, Type-2, or Type-3. The TAG numbers are enclosed in square brackets.

```
Type      ::=  CHOICE  {      [1]      Type-1,
                           [2]      Type-2,
                           [3]      Type-3
                     }
```

SEQUENCE { } - is the cumulative indicator in a production definition. It can be read as "and". A sequence of data entities are separated by commas within the brackets of the SEQUENCE indicator. For example, a data entity process consists of the ordered sequence of entities Step-1, Step-2, Step-3. Explicit TAGs are not required since the order of the sequence is fixed.

```
Process  ::=  SEQUENCE  {      Step-1,
                               Step-2,
                               Step-3
                     }
```

SET { } - is also a cumulative indicator in a production definition, which can also be read as "and". It is similar to the SEQUENCE indicator except that the order of the elements is not fixed. Therefore TAG codes must be used to distinguish the elements. For example, data entity Flock consists of the unordered sequence of entities Bird-1, Bird-2, Bird-3.

```
Flock      ::=  SET  {      [1]      Bird-1,
                           [2]      Bird-2,
                           [3]      Bird-3
                     }
```

SEQUENCE OF - is a series indicator in a production definition. It permits zero or more data entities of the same type to be part of a production. The end of the sequence is determined when there are no more data elements of the same type. For example, a data element List is composed of a series of Entries (Entry-1, Entry-2, Entry-3,...).

```
List      ::=  SEQUENCE OF  Entries
```

Data entities in a SEQUENCE or a SET may be optional. Optional data elements are indicated by the keyword OPTIONAL after the data element name. A default value may also be defined for an optional data element. This is indicated by the keyword DEFAULT after the data element name followed by the value of the default state. The keyword DEFAULT implies the keyword OPTIONAL. For example, data entity Group consists of the ordered sequence of entities Element-1, Element-2, Element-3, Element-4, where Element-2 is optional and Element-3 is optional and takes on the default value 1. Explicit TAGs are required in order to distinguish which elements are included.

```
Group      ::=  SEQUENCE  {  [1]  Element-1,
                           [2]  Element-2 OPTIONAL,
                           [3]  Element-3 DEFAULT (1),
                           [4]  Element-4
                     }
```

-- signifies the beginning of comments that are not part of the grammar, but are included to clarify the semantic meaning and context.

Keywords - such as SEQUENCE, SET, CHOICE etc. are expressed in capital letters.

identifier name - the name of an identifier consists of a string of letters, digits and hyphens beginning with a lower case letter. Identifier names may be used to identify particular elements of an entity or to assign a value to a particular element of an entity.

Entity Name - the name of an entity consists of a string of letters, digits and hyphens beginning with an upper case letter.

The primitive level of the Abstract Syntax Notation consists of a number of built-in data types. These can be classified into several categories as shown below:

Numerical Data Types	- BOOLEAN INTEGER	- True or False - Integer number (a signed integer number of arbitrary length)
Bit Oriented Data Types	- BIT STRING OCTET STRING	- String of bits - String of 8-bit bytes
General Data Types	- NULL ANY	- Null string - Any data string
Time/Date Data Type	- GENERALIZED TIME	- Date / Time
Character Data Types	- GRAPHIC STRING	- A string of characters

These primitive data types correspond to those defined in ISO 19103 and used in the models in The ISO TC211 suite of standards.

This description of the Meta Notation only identifies the principal functions of ASN.1. A more comprehensive description is given in the ISO standard ISO 8824.

Appendix D. Profile for Deposit with the UN

(Normative)

The Explicit Text Format provides a general set of structures that can express any of the elements in an S-121 data base with the limitation that only one Metadata, Governance, and Basic Administrative Unit object may be expressed in one data set. Nations that wish to represent their proclamations, decrees or laws in the Explicit Text Format have the capability to do so. However, the requirements for deposit with the UN/DOALOS are much simpler.

This profile provides a guide to foster consistency where desired. Some information types such as Spaces are not included in the profile. Only a limited number of attributes are included.

ISO and S-100 require Metadata to be included with any dataset. In the Explicit Text Format, Metadata forms a “cover page” that may precede an Explicit Text Formatted document. This Metadata is optional and where used only a minimum of metadata is needed.

The names of attributes used in the Record Type Identifier in the Explicit Text Format derive from the attribute names in the S-121 model. These names may be replaced by alternate names (aliases) that are defined explicitly in this profile in order to enhance readability. These alternate names are given in the tables below.

D 1. Profile File Structure

A data set for deposit consists of the following blocks where applicable:

- Metadata Block
- Governance Block
- Party Block Group
 - Party Block ...
- Basic Administrative Block
- Rights Block Group
 - Rights Block ...
- Responsibilities Block Group
 - Responsibilities Block ...
- Restrictions Block Group
 - Restrictions Block ...
- Source Block Group
 - Source Block ...
- Zone Block Group
 - Zone Block ...
- Limit Block Group
 - Limit Block ...
- Curve Block Group
 - Curve Block ...
- Location Block Group
 - Location Block ...
- Point Block Group
 - Point Block ...

D 2. Profile Metadata Block

A subset of the available metadata fields are selected for the deposit profile. Since the profile is only a recommendation other metadata may be used as needed. There is only one Metadata Block per data file so there is no Block ID field in the Metadata Block Descriptor Record.

Profile Metadata Block First Record:

Metadata Block Descriptor Record Contents	
Metadata	

Subsequent Records:

Recommended Name	Attribute Name	Recommended content
Title:	title	Name of deposit
Abstract:	abstract	Brief narrative summary of the dataset's contents
Topic:	topic	Topic areas boundaries, and oceans.
Nation:	metadataContactOrganisation	Nation making the deposit and therefore responsible for the metadata
Date:	referenceDate	reference date for the deposit
Language:	datasetLanguage	Languages of the dataset using standard ISO 639-2 three letter codes. Note: Three letter language code followed by an optional ISO 3166 three letter country code, For example ENG for English and FRA for French
Metadata valid on:	metadataDate	Metadata creation date
File:	metadataFileIdentifier	A unique phrase or string which uniquely identifies the metadata file
Date Type:	datasetReferenceDateType	Type of date from code list: - Creation Date - Publication Date - Revision Date - Date Became In Force - Date Became Adopted
West Extent:	westLongitudeExtent	West limit of extent.
East Extent:	eastLongitudeExtent	East limit of extent.
South Extent:	southLatitudeExtent	South limit of extent.
North Extent:	northLatitudeExtent	North limit of extent.

D 3. Profile Governance Block

The profile of the Governance Block for deposit is a subset of the available elements. The names are selected to be meaningful for deposit.

The Description Attribute is expected to be many lines of text that will make use of the Record Extension capability to allow multiple paragraphs. This will correspond to the preamble of the deposit.

There is only one Governance Block per data file so there is no Block ID field in the Governance Block Descriptor Record. Reference to the Governance block from the Basic Administrative Unit Block is assumed.

Profile Governance Block First Record:

Governance Block Descriptor Record Contents	
Maritime Limits and Boundaries deposit	

Subsequent Records:

Recommended Name	Attribute Name	Recommended content
Reference Number:	referenceNumber	Legislative reference number
Title:	governanceTitle	Title of the legislative reference. This is the general title of the whole deposit
Description:	governanceDescription	Description of the governance statement. This can be a multi paragraph preamble to the deposit
Approval Date:	dateApproved	Date at which the legislative statement or document was approved by the appropriate governing body
Validity Date:	startLifespan	Reference date for the validity of this version of the dataset. This is the statement of versioning for an entire dataset. Note that the Explicit Text Format is versioned by dataset. If Metadata is provided then this date is equivalent to Reference Date
Collection:	collection	The State name should be used when depositing

D 4. Profile Party Block Group

In a deposit there normally is only one party involved. The state makes a declaration of its position.

The Party Block Group consists of two blocks. The first block in the group contains only a Group Block Descriptor Record. This allows a record of descriptive text to be expressed. The second block describes the party. In the case where there is only one party the use of the Party Group Block Descriptor Record is optional; however, it makes the output more readable.

An example of both options are shown below. The first option uses “name” as the Block ID, whereas the second option uses an object id (Oid) as the Block ID.

Party: Australia
Type: State Country

Party: Party-2
Party Name: New Zealand
Type: State Country

Both are valid and can be used together; however, the first option is more readable.

Profile Party Block Group Description:

Party Group Block Descriptor Record Contents
The information in this file relates to the following entity(s)

Profile Party Block First Record (multiple Party Blocks may exist where applicable):

Party Block Descriptor Record Contents	Block ID
Party:	Identifier of the Party block in the Oid format for example “Party-1” or a unique “name”. In the case of “Party” the unique name would derive from the partyName attribute. Therefore the partyName attribute is not needed.

Subsequent Records

Recommended Name	Attribute Name	Recommended content
Party Name:	partyName	The name of the party
Type:	partyType	The type of the party - "State Country" stateCountry
Group:	groupPartyType	The type of the group party: - agreement - association - condominium
Individuality:	isIndividualOrGroupParty	- individual "individualParty" - group "groupParty"

D 5. Profile Basic Administrative Unit Block

For a deposit the official entity is represented by the Basic Administrative Unit. Only one Basic Administrative Unit is permitted in one data file in the Explicit Text Format. All references to the Basic Administrative Unit from the Feature unit (Zone, Limit and Location) are assumed. Therefore there is no need to express a Block ID for the Basic Administrative Unit.

It is necessary for the reader to know what type of entity is being described. This can be done using the Administrative Unit Name. This makes the Administrative Unit Name mandatory in the profile.

Profile BAUnit Block First Record:

BAUnit Block Descriptor Record Contents
This file depicts the following object

Subsequent Records:

Recommended Name	Attribute Name	Recommended content
Administrative Unit Name:	basicAdministrativeUnitName	Name for an instance of a "BasicAdministrativeUnit" (for example Continental Shelf)
Context:	basicAdministrativeUnitContext	Description of the context for an instance of a "BasicAdministrativeUnit"
Right:	right	Reference to Right object
Responsibility:	responsibility	Reference to Responsibility object
Restriction:	restriction	Reference to Restriction object
Party:	party	Reference to Party object
Source:	source	Reference to Source object

D 6. Profile Right Restriction and Responsibility Blocks

D 6.1. Profile Rights Block

For a deposit, only a subset of the description of a Right description is required.

The Right Block Group consists of two or more blocks. The first block in the group contains only a Group Block Descriptor Record. This allows a record of descriptive text to be expressed. The second and subsequent blocks describe the Right(s). The Block ID for a Right Block may be either the object identifier (Oid) or a name. The name is taken from the Right Type attribute. If a name is used as the Block ID then the Right Type attribute is not required.

An example of both options are shown below. The first option uses “name” as the Block ID, whereas the second option uses an object id (Oid) as the Block ID.

Right: Sovereign Right
Description: Australia Sovereign Right
Party: Australia

Right: Right-3
Right Type: Access Right
Description: Australia Access Right
Party: Australia

Both are valid and can be used together; however, the first option is more readable.

This is also true in the case of the Restriction and Responsibility Blocks.

Profile Rights Block Group Description:

Rights Group Descriptor Record Contents	
Which relate to the following rights	

Profile Rights Block First Record (multiple Rights Blocks may exist):

Rights Block Descriptor Record Contents	Block ID
Right:	Identifier of the Rights block in the Oid format for example “Right-1” or a unique “name”. In the case of “Right” the unique name may derive from the rightType attribute (for example Easement Right). In this case the rightType attribute is not needed

Subsequent Records:

Recommended Name	Attribute Name	Recommended content
Right Type:	rightType	The type of the right: Character String
Description:	description	Description of the right.
Party:	party	Reference to Party object
Source:	source	Reference to Source object

D 6.2. Profile Restrictions Block

For a deposit the only a subset of the description of a Restriction description is required.

Profile Restriction Block Group Description:

Restriction Block Descriptor Record Contents	
Which relate to the following restrictions	

Profile Restriction Block First Record (multiple Restrictions Blocks may exist):

Restriction Block Descriptor Record Contents	Block ID
Restriction:	Identifier of the Restrictions block in the Oid format for example "Restriction-1" or a unique "name". In the case of "Restriction" the unique name may derive from the restrictionType attribute (for example Use Restriction). In this case the restrictionType attribute is not needed

Subsequent Records:

Recommended Name	Attribute Name	Recommended content
Restriction Type:	restrictionType	The type of the Restriction: Character String
Description:	description	Description of the restriction
Party Required:	partyRequired	Indicates whether a party is required for the registration of the restriction in the association to Party
Party:	party	Reference to Party object
Source:	source	Reference to Source object

D 6.3. Responsibility Block

An S121_Responsibility is a formal or informal obligation to do something. The ResponsibilityBlock describes a responsibility.

For a deposit the only a subset of the description of a Restriction description is required.

Profile Responsibility Block Group Description:

Responsibilities Block Descriptor Record Contents
Which relate to the following responsibilities

Profile Responsibility Block First Record (multiple Responsibility Blocks may exist):

Responsibilities Block Descriptor Record Contents	Block ID
Responsibility:	Identifier of the Responsibilities block in the Oid format for example "Responsibility-1" or a unique "name". In the case of "Responsibility" the unique name may derive from the responsibilityType attribute (for example Maintenance Responsibility). In that case the responsibilityType attribute is not needed

Subsequent Records:

Recommended Name	Attribute Name	Recommended content
Responsibility Type:	responsibilityType	The type of the Responsibility: Maintenance Responsibility: Character String
Description:	description	Description of the responsibility
Party:	party	Reference to Party object
Source:	source	Reference to Source object

D 7. Profile Source Block

All of the information elements in the S121 model may have sources. In a deposit the source is the depositing nation so many of the source attributes are not required.

Profile Source Block Group Description:

Source Block Descriptor Record Contents	
The source(s) of this object(s) is	

Profile Source Block First Record (multiple Source Blocks may exist):

Source Block Descriptor Record Contents	Block ID
Source:	Identifier of the Source block in the Oid format for example "Source-1"

Subsequent Records:

Recommended Name	Attribute Name	Recommended content
Source Name:	sourceDocumentName	Document name - for example the document (legislation, treaty, title) that defines the object
Online Source:	onlineResourceLinkageURL	official URL (or equivalent online resource) where the document is distributed;.Multiple online resource URL may be included in one character string separated by commas
Source Registry Number:	sourceRegistryNumber	Unique official identifier of the record in a registry. For example, in states with registers of legislative instruments, versioning is controlled by the registry ID
Source Date:	sourceAuthoritativeDate	The date of force of law of the source by an authority
Responsible Organization:	responsiblePartyOrganizationName	Name of the responsible organization
Role:	responsiblePartyRole	Role of the responsible party from code list: - Resource Provider resourceProvider - Custodian custodian - Owner owner - User, user - Distributor, distributor - Originator, originator - Point Of Contact, pointOfContact - Principle Investigator, principleInvestigator - Processor, processor - Publisher, publisher - Author. author

D 8. Profile Zone Block and Surface Block Group

Zones with associated geometry may be part of a deposit.

In the S-121 model the geometry is a component of the feature object. This means that a Zone feature may be followed directly by the associated surface geometry. Typically zones are defined by reference to the limits that bound the zone and not to a specific surface geometric element. Therefore the Surface Block would be seldom expressed in the Explicit Text Format unless there was a need to express the Descriptive Location (Location by Test), Reference System, attributes or a Source reference attribute.

Profile Zone Block and Surface Block Group Description

Zone and Surface Group Block Descriptor Record Contents
The extent of the object is defined by the following zones

D 8.1. Profile Zone Block

All of the available attributes associated with Zones from the standard are made available in the profile but only those that are required need be chosen.

Profile Zone Block First Record (multiple Zone Blocks may exist):

Zone Block Descriptor Record Contents	Block ID
Zone Identifier:	Identifier of the Zone block in the Oid format for example "Zone-1"

Subsequent Records:

Recommended Name	Attribute Name	Recommended content
Label:	label	A short textual identifier of the feature unit
Name:	name	The name of the feature
Context:	context	The administrative aspects of the feature object
Releasability:	releasabilityType	This attribute is optionally used to differentiate between releasability status for particular features: - Official, official, - Internal, internal, - Controlled controlled
Zone Object Type:	zoneObjectType	This attribute identifies which Feature Unit Limit type this object refers to based on the Feature Catalogue: - Internal Water internalWater - Archipelagic Waters archipelagicWaters - Territorial Sea territorialSea - Exclusive Economic Zone exclusiveEconomicZone - Contiguous Zone contiguousZone - Continental Shelf continentalShelf - Roadsteads roadsteads - High Seas highSeas - The Area theArea
Area Value:	areaValue	This attribute describes the value of the area of the zone
Jurisdiction Domain Type:	jurisdictionDomainType	Defines the vertical juridical domain of the object: - Airspace airspace - Land Surface landSurface - Water Surface waterSurface - Water Column waterColumn - Seabed Surface seabedSurface - Subsoil subsoil
Area Type:	areaType	This attribute describes the type of the area of the zone: - Official Area officialArea - Nonofficial Area nonofficialArea - Calculated Area calculatedArea - Surveyed Area surveyedArea
Area Surface Relation:	areaSurfaceRelation	Relationship to surface: - Mixed mixed - Below below - Above above - On Surface onSurface
Area Value:	areaValue	This attribute describes the value of the area of the zone
Area Unit Of Measure:	areaUnitOfMeasure	The unit of measure used to define the area of the Zone
Surface:	surface	Reference to the associated surface geometric object
Bounded By:	boundary	Reference to limit or limits that define the outer boundary of zone
Inner Boundary:	innerBoundary	Reference to limit or limits that define one or more inner boundary of zone (holes such as for islands)

Recommended Name	Attribute Name	Recommended content
Administrative Unit:	administrativeUnit	Reference to the Basic Administrative Unit to which this feature is associated
Source:	source	Reference to the source object

D 8.2. Profile Surface Block

The Surface block is an object that defines the spatial geometry of a Surface. The geometry of a surface is normally implemented by the use of pointers to the bounding Limits. The object S121_Surface carries attributes about the underlying geometric surface but does not explicitly define a geometric surface. The Surface Block would be seldom expressed in the Explicit Text Format unless there was a need to express any of the attributes (Descriptive Location, Reference System, or Source).

Profile Surface Block First Record (multiple Surface Blocks may exist):

Curve Block Descriptor Record Contents	Block ID
Surface:	Identifier of the Surface block in the Oid format (used with individual blocks) for example "Surface-1"

Subsequent Records:

Recommended Name	Attribute Name	Recommended content
Descriptive Location:	locationByText	This attribute allows a spatial attribute to be a textual description. This allows "Location", "Limit", "Zone"s and "Space" that are not fully described geometrically to be included Quantity Specifications
Reference System:	referenceSystem	Allows a CoordinateReferencingSystem (CRS) to optionally be specified. In many other S-100 based products the CRS is only defined at the metadata level and applies for the whole data set; however, in S-121 it is necessary to detail it right down to the specific instances of geometry since treaty points and lines may come from different sources such as different treaties that may use different CRS
Source:	source	Reference to the source object

D 9. Space Blocks

Spaces are not part of the deposit and are not included in this profile.

D 10. Profile Limit Block and Curve Block Group

Outer limits and other limits and boundaries with associated Curve geometry may be part of a deposit.

A Limit feature may be followed directly by the associated Curve geometry in the same group because geometry is a component of a feature. This grouping makes the printed output more readable; however, referencing between limits and curves is permitted to allow for shared geometry.

Profile Limit and Curve Block Group Description:

Limit and Curve Group Block Descriptor Record Contents
The outer limits are defined by the following limits and associated curves

Note a Record Extension record can be used to extend the Limit and Curve Group Block Descriptor line to add additional text to make the data more readable. For example the following shows a Limit and Curve Group Block Descriptor Record with an extension record.

The outer limits are defined by the following lines and associated curves

The outer limit of the Continental Shelf related to Macquarie Island (Zone-7) area is defined by the following lines depicted in their originally declared datum

D 10.1. Profile Limit Block

Outer limits and other limits and boundaries may be part of a deposit. All of the available attributes from the standard are made available in the profile but only those that are required need be chosen.

The Delimiting Points attribute is optional. It carries a pointer between a limit object and an associated Location such as a Baseline Point. This pointer would be used for specific named Location objects; however, if this pointer structure were to be used for all relations between limits and associated locations there may be too many pointers and the explicit text output would become unreadable. An alternative reference that implements the relation between limits and associated points is described in the Location and Point Block section that groups pointers and is therefore more readable.

Profile Limit Block First Record (multiple Limit Blocks may exist):

Limit Block Descriptor Record Contents	Block ID
Limit Identifier:	Identifier of the Limit block in the Oid format for example "Limit-1"

Subsequent Records:

Recommended Name	Attribute Name	Recommended content
Label:	label	A short textual identifier of the feature unit
Name:	name	The name of the feature
Context:	context	The administrative aspects of the feature object
Releasability:	releasabilityType	This attribute is optionally used to differentiate between releasability status for particular features: - Official official - Internal internal - Controlled controlled
Limit object type:	limitObjectType	This attribute identifies which Feature Unit Limit type this object refers to based on the Feature Catalogue: - Boundary boundary - International Boundary internationalBoundary - Baseline baseline - Normal Baseline .normalBaseline - Straight Baseline straightBaseline - Archipelagic Baseline archipelagicBaseline - Low Tide Elevations lowTideElevationsBaseline - Mouths of Rivers mouthsOfRiversBaseline - Reefs ReefsBaseline - Bays baysBaseline - Ports portsBaseline - Outer Limit of the Territorial Sea outerLimitOfTheTerritorialSea - Outer Limit of the Contiguous Zone outerLimitOfTheContiguousZone - Outer Limit of the Exclusive Economic Zone outerLimitOfTheExclusive Economic Zone - Outer Limit of the Continental Shelf outerLimitOfTheContinentalShelf - Outer Limit of the Roadstead outerLimitOfTheRoadstead - Construction Line constructionLine
Type of Arc:	arcGeometryType	Type of computation used to define an arc (line): - Geodesic geodesic, - Loxodrome loxodrome.

Recommended Name	Attribute Name	Recommended content
Administrative Unit:	administrativeUnit	Reference to the Basic Administrative Unit to which this feature is associated
Source:	source	Reference to the source object
Curve:	curve	Reference to the associated curve geometric object
Delimiting Points:	delimitingPoints	Reference to the beginning and end points that delimit a limit

D 10.2. Profile Curve Block

Curves provide the geometry for a limit. A set of vertices describing a limit is best described in a table as long as the number of attributes expressed is not excessive. Note that if an attribute is common for all rows of a table it can be expressed in the descriptor record at the head of the table. If Location by Text is used then individual Curve Blocks should be used.

Profile Curve Block First Record (multiple Curve Blocks may exist or a table may be used):

Curve Block Descriptor Record Contents	Block ID
The limit is defined by the following curve:	Identifier of the Curve block in the Oid format (used with individual blocks) for example "Curve-1". This is derived from the attribute "saID" the identifier of the entire curve

Subsequent Records:

Recommended Name	Attribute Name	Recommended content
Vertex Identifier:		A generated identifier for each row in a table. This identifies each vertex in a table row
Descriptive Location:	locationByText	This attribute allows a spatial attribute to be a textual description. This allows "Location", "Limit", "Zone"s and "Space" that are not fully described geometrically to be included Quantity Specifications
Datum:	referenceSystem	Allows a CoordinateReferencingSystem (CRS) to optionally be specified. In many other S-100 based products the CRS is only defined at the metadata level and applies for the whole data set; however, in S-121 it is necessary to detail it right down to the specific instances of geometry since treaty points and lines may come from different sources such as different treaties that may use different CRS
Curve Geometry:	curveGeometry	Polyline or arcByCentrePoint or circleByCentrePoint formatted character string consisting of: <ul style="list-style-type: none"> - Sequence of vertices (longitude coordinate, latitude coordinate or Direct Position), or - arcByCentre (arcStart longitude, slash delimiter, arcStart latitude, comma delimiter, arcEnd longitude, slash delimiter, arcEnd latitude, comma delimiter, arcCentre longitude, slash delimiter, arcCentre latitude), or - circleByCentre (circleCentre longitude, slash delimiter, circleCentre latitude, comma delimiter, circlePoint longitude, slash delimiter, circlePoint latitude)

Recommended Name	Attribute Name	Recommended content
WGS84Curve Geometry:	WGS84CurveGeometry	Polyline or arcByCentrePoint or circleByCentrePoint formatted character string consisting of: <ul style="list-style-type: none"> - Sequence of vertices (longitude coordinate slash delimiter latitude coordinate), or - arcByCentre (arcStart longitude, slash delimiter, arcStart latitude, comma delimiter, arcEnd longitude, slash delimiter, arcEnd latitude, comma delimiter, arcCentre longitude, slash delimiter, arcCentre latitude), or - circleByCentre (circleCentre longitude, slash delimiter, circleCentre latitude, comma delimiter, circlePoint longitude, slash delimiter, circlePoint latitude)
Source:	source	Reference to the source object

D 11. Profile Location and Point Blocks

The Location object provides information about a specific position. The Point geometry object provides its geographic Latitude / Longitude position. A Location object can only have one Point geometry object associated with it, but a point geometry object can be shared by more than one Location object. That is, physically more than one thing can exist at the same Latitude / Longitude position.

Normally there is a direct one to one relation between a Location object and the associated Point geometry object; however, Point geometry objects may be shared.

Location objects with their associated point geometry read best as a single table. If there is a one to one relation between the Location objects and the associated points then both can be included in a single row of a combined table. If there is no point sharing then there is no need to have an identifier on the point geometry objects. If point sharing occurs then there needs to be a separate Location object block and a Point object block table.

In producing readable text the width of a table is a critical issue. There is only room for four or five columns. If there are too many attributes expressed then separate tables for the Location object and Point object are also required. One approach to reducing the number of attribute columns that need to be included in a table is to attach simple attributes to the whole Group for those attributes that do not change in the table. This means adding simple records immediately after the Group Block Description record.

Profile Location Block and Point Block Group Description:

Location and Point Block Descriptor Record Contents
The identified locations and associated points are given below

D 11.1. Profile Location Block

This block is used to describe Location objects that reference Point objects. This allows for Point geometry sharing.

All of the available attributes from the standard are made available in the profile but only those that are required need be chosen.

Profile Location Block First Record (multiple Location Blocks may exist):

Location Block Descriptor Record Contents	Block ID
Location:	Identifier of the Location block in the Oid format for example "Location-1"

Subsequent Records:

Recommended Name	Attribute Name	Recommended content
Location Identifier:	fulD	Identifier of the Location object feature unit in the "Oid" (Object ID) format. The "Oid" comprises a unique character string and a namespace identifier which is also a unique character string. An example might be "Location-1" where "Location" is the namespace and "1" is the unique id within that namespace
Label:	label	A short textual identifier of the feature unit
Name:	name	The name of the feature
Context:	context	The administrative aspects of the feature object
Releasability:	releasabilityType	This attribute is optionally used to differentiate between releasability status for particular features: <ul style="list-style-type: none"> - Official official - Internal internal - Controlled controlled
Type:	locationObjectType	This attribute identifies which Feature Unit Location Object type this object refers to based on the feature catalogue: <ul style="list-style-type: none"> - Baseline Point baselinePoint - Contributing Point contributingPoint - Limit Point limitPoint - Boundary Point boundaryPoint
Interpolation:	interpolationRole	The role of point in the structure of a "Straight" line or curve <ul style="list-style-type: none"> - Deflection deflection - Densification densification
Point Type:	pointType	The type of point: <ul style="list-style-type: none"> - Defined defined - Computed computed
Administrative Unit:	BAUnit	Reference to the Basic Administrative Unit to which this feature is associated
Source:	source	Reference to the source object
Point:	point	Reference to the associated point geometric object

D 11.2. Profile Point Block

Lists of points provide the geometry for a set of Locations. A list of points is best described in a table as long as the number of attributes expressed is not excessive. If Location by Text is used then individual Point Blocks rather than a table should be used.

Profile Point Block First Record (multiple Point Blocks may exist or a table may be used):

Point Block Descriptor Record Contents	Block ID
Point:	Identifier of the Point block in the Oid format (used with individual blocks) for example "Point-1"

Subsequent Records:

Recommended Name	Attribute Name	Recommended content
Point Identifier:	saID	The spatial attribute identifier in the Oid format (used in a table)
Descriptive Location:	locationByText	This attribute allows a spatial attribute to be a textual description. This allows "Location", "Limit", "Zone"s and "Space" that are not fully described geometrically to be included Quantity Specifications

Recommended Name	Attribute Name	Recommended content
Datum:	referenceSystem	Allows a CoordinateReferencingSystem (CRS) to optionally be specified. In many other S-100 based products the CRS is only defined at the metadata level and applies for the whole data set; however, in S-121 it is necessary to detail it right down to the specific instances of geometry since treaty points and lines may come from different sources such as different treaties that may use different CRS
Original Location:	originalLocationTextualPosition	Reference location of a point when it was first established
Original Location Reference System:	originalLocationReferenceSystem	Reference System for a Reference location of a point when it was first established
Official Location:	officialLocationTextualPosition	Official location reference of the point
Official Location Reference System:	officialLocationReferenceSystem	Reference System for an official reference of the point
Global Location	globalLocationTextualPosition	Global reference to a point in a world global datum such WGS84 or ITRF2000
Global Location Reference System:	globalLocationReferenceSystem	Reference System for a global reference to a point in a world global datum such WGS84 or ITRF2000
WGS84 Location:	WGS84LocationTextualPosition	WGS84LocationTextualPosition
Common Location:	commonLocationTextualPosition	Common reference location of point in a datum that is defaulted to the other points in the dataset
Common Location Reference System:	commonLocationReferenceSystem	Reference System for a common reference location of point in a datum that is defaulted to the other points in the dataset
Points:	pointGeometry	GM_Point geometry as per IHO S-100 Part 7 Spatial Schema. The associated data type is structured as a formatted Character String to assist in implementation; however, a GIS system encode a Direct Position as an internal numeric type and the type can be converted to a character string upon encoding
Source:	source	Reference to the source object

D 12. Combined Location Object and Point Object Block

When there is a one to one relationship between Location objects and the associated Point objects, both can be represented in a single table. This is a common situation.

The Block ID field of the Descriptor Record contains an identifier of the associated Limit object. This allows one to not use the Delimiting Points attribute (from the Limit object), making the explicit text output more readable. This ID is structured as the word from: followed by the ID of the associated Limit object.

Profile Location Object and Point Object Block First Record:

Combined Location and Point Block Descriptor Record Contents					
Locations and associated points:				from: ID of associated Limit object	
Record Type	Record Type	Record Type	Table Header Row Record Type	Record Type	Record Type
Record Content	Record Content	Record Content	Table Row Record Content	Record Content	Record Content
...					

For example a table of Location and associated Point geometry objects might be as follows.

The identified locations and associated points are given below.

The outer limits are defined by the following (Original) locations.

Type: Limit Point

Interpolation: Densification.

Locations and associated points: from: Limit-462

Macquarie Island (Zone-7)

Point Identifier	Latitude	Longitude	Datum
MAC-CS-1	-51°04'48.9600"	158°01'25.9800"	ITRF2000
MAC-CS-25	-57°21'25.1880"	163°23'44.0250"	ITRF2000
MAC-CS-26	-57°21'14.3454"	163°23'19.7500"	ITRF2000

Page intentionally left blank

Appendix E. Record Oriented Encoding Structure in Abstract Syntax Notation ASN.1

(Normative)

This Appendix describes the structure of the Explicit Text Record Oriented Format using the Abstract Syntax Notation (ASN.1). A description of the ASN.1 description language is given in Appendix B.

The metadata elements are derived from the S-121 Product Specification that themselves are derived from S-100. Additional metadata may be added in accordance with S-100 and ISO 19115 Geographic Information Metadata.

S-121 Data Set Abstract Syntax

```
-- ----- Dataset Structure ---
-- An S121 Data set consists of a Sequence of Blocks corresponding to each type of information element
-- The first two blocks are MetadataBlock, and a GovernanceBlock. For simplicity there may only
-- be one MetadataBlock, GovernanceBlock and BAUnitBlock per dataset. These blocks are required, but they may
-- be empty; that is, a block may consist of only an EndOfBlockMarker.

S121DataSet
  ::= SEQUENCE {
    MetadataBlock,
    GovernanceBlock,
    SEQUENCE OF
      CHOICE {
        Block,
        Group
      }
    EndOfFile
  }

-- ----- Block Structure ---
-- Each S121 Block corresponds to an Information Element class in the S-121 UML model. There are two types of
blocks, Linear Blocks and Table Blocks.

Block
  ::= CHOICE{
    LinearBlock
    TableBlock
  }

-- Each S121 LinearBlock consists of a set of Records terminated by an end of EndOfBlockMarker.
-- An EndofBlockMarker is equivalent to a blank record at the end of a block resulting in two
-- EndOfRecordMarkers in a row.

LinearBlock
  ::= SEQUENCE OF{
    BlockDescriptorRecord
    SEQUENCE {Record }
    EndOfBlockMarker
  }

-- ----- Record Structure ---
-- An S121 Record contains the
-- information pertaining to an
-- information element from the
-- S121 model.

Record
  ::= CHOICE{
    TextRecord
    BinaryRecord
  }

-- ----- Text Record Structure ---
-- An S121 TextRecord contains the
-- information pertaining to an
-- information element from the
-- S121 model.

TextRecord
  ::= SEQUENCE {
    TextType
    TextContent
  }

-- ----- Binary Record Structure ---
-- An S121 BinaryRecord contains the
-- information pertaining to an
-- information element from the
-- S121 model.

BinaryRecord
  ::= SEQUENCE {
    BinaryType
    BinaryContent
  }
```

```

-- Each S121 TableBlock consists of a record header row describing attribute type names for columns and
-- a set of data rows with attributes.

TableBlock
    ::= SEQUENCE OF{
        BlockDescriptorRecord
            -- An S121 TableBlock contains the
            -- information pertaining to a set of
            -- information element of the same type
            -- organized in a table.

        TableHeaderRow
            -- An S121 BlockDescriptorRecord
            -- identifies the type of information
            -- element related to the block.
            -- An TableHeaderRow describes the names
            -- of attribute value types
            -- A TableRow describes the values
            -- corresponding to the attribute types
            -- for each column.
            -- A block is terminated by an
            -- EndOfBlockMarker which is a
            -- blank record resulting in two
            -- EndOfRecordMarkers in a row

    }

BlockDescriptorRecord
    ::= SEQUENCE {
        BlockTypeIdentifier,
            -- Character string describing the
            -- the type of information
            -- element related to the block.
            -- (IA5 Code 09)
            -- This identifier is used when it is
            -- necessary to reference a block
            -- (corresponding to an information element)
            -- Record termination delimiter sequence

        TAB_Delimiter
        BlockID
            -- For a Party Block or a Source Block this can
            -- be a unique name of the Party or Block taken
            -- from the name attribute in that element
            -- Object Identifier. This identifier is a
            -- formatted character string consisting of a
            -- namespace followed by a dash followed by a
            -- number. The namespace is derived from the
            -- name of the object such as "Right".

        EndOfRecordMarker
    }
    BlockTypeIdentifier      ::= Graphic String

BlockID
    ::= CHOICE {
        NameAsID,
            -- Unique character string corresponding to a
            -- name taken from the "sourceDocumentName"
            -- attribute of a Source object or the
            -- "partyName" attribute of a Party object or
            -- from other unique names associated with an
            -- object.

        Oid_FormattedCharacterString
    }

NameAsID
    ::= Graphic String
    -- For a Party Block or a Source Block this can
    -- be a unique name of the Party or Block taken
    -- from the name attribute in that element
    -- Object Identifier. This identifier is a
    -- formatted character string consisting of a
    -- namespace followed by a dash followed by a
    -- number. The namespace is derived from the
    -- name of the object such as "Right".

-- ----- Group Structure ---
-- Blocks may be grouped. That is a set of blocks of the same type may be expressed sequentially. At the
-- beginning of the group there may be a block that consists of only a BlockDescriptorRecord that expresses
-- text that described the group to the human reader. Grouping is not required but it makes the explicit text
-- output more readable and allows for these additional descriptor records to be expressed improving
-- readability.

-- A Group consists of a number of blocks. The group is identified by a BlockDescriptorRecord. This Block
-- descriptor record may be followed by a sequence of records that define attributes that are valid for the
-- whole group. This allows constant values such as a source to be expressed once for the group. The group then
-- consists of a sequence of blocks. A group is terminated by a blank block, that is by the terminator CRLF
-- CRLF following the end of a block, or specifically CRLF CRLF CRLF CRLF.

Group
    ::= SEQUENCE OF{
        BlockDescriptorRecord
            -- An Group is a sequence of blocks.

        SEQUENCE {Record }
            -- A Group begins with a BlockDescriptorRecord
            -- which identifies the type of group of blocks
            -- An sequence of S121 Record contains the
            -- record values that apply to the whole group.
            -- An sequence of Blocks compose a group
            -- A Group is terminated by an EndOfBlockMarker
            -- which is a blank record resulting in two
            -- EndOfRecordMarkers in a row. Since this is

        SEQUENCE {Block }
        EndOfBlockMarker
    }

```

```

        -- after the last block in the group this is
        -- composed of specifically
        -- CRLF CRLF CRLF CRLF .
    }

-- ---- Record Structure ---

-- Each S121 Record consists of a S121 RecordTypeIdentifier followed by a TAB_Delimiter character
-- followed by a S121_RecordContents and terminated by an EndOfRecordMarker. A record contains the information
-- pertaining to an attribute type and value. The type of attribute is carried in the RecordTypeIdentifier and
-- the value is carried in the RecordElement. If a record contains a blank RecordTypeIdentifier; that is, the
-- record starts with the Tab_Delimiter, then the RecordElement contents is considered to be an extension of
-- the previous record. In this way attribute values, such as character strings that contain CRLF or Cr or LF
-- characters as part of the attribute can be supported.

Record
 ::= SEQUENCE {
    RecordTypeIdentifier,                                -- Unique identifier of the record type
    TAB_Delimiter,                                     -- (IA5 Code 09)
    RecordContent,                                     -- Record contents which may be multiple
                                                       -- elements separated by commas.
    EndOfRecordMarker                                 -- Record termination delimiter sequence
}

RecordContent
 ::= SEQUENCE {
    RecordElement,                                     -- Character string of the data value
    OPTIONAL SEQUENCE OF SET { COMMA_Delimiter ,      -- Subsequent record elements
                               SPACE OPTIONAL,           -- separated by COMMA characters (IA5 Code 32)
                               RecordElement } ,          -- and an optional SPACE character (IA5 Code44)
    EndOfRecordMarker                                -- Record termination delimiter sequence
}

-- ---- Table Structure ---

-- Each TableHeaderRow consists of a set of RecordTypeIdentifier(s) seperated by TAB_Delimiter characters

TableHeaderRow
 ::= SEQUENCE {
    RecordTypeIdentifier,                                -- Unique identifier of the record type in
                                                       -- the first table column
    OPTIONAL SEQUENCE OF SET { TAB_Delimiter,           -- Subsequent record type identifiers
                               RecordTypeIdentifier } ,   -- separated by TAB characters (IA5 Code 09)
    EndOfRecordMarker                                 -- Record termination delimiter sequence
}

-- Each TableRow consists of a set of RecordElement(s) seperated by TAB_Delimiter characters

TableRow
 ::= SEQUENCE {
    RecordElement,                                     -- Character string of the data value
                                                       -- of the the first table column
    OPTIONAL SEQUENCE OF SET { TAB_Delimiter ,         -- Subsequent record elements in columns
                               RecordElement } ,       -- separated by TAB characters (IA5 Code09)
    EndOfRecordMarker                                -- Record termination delimiter sequence
}

RecordTypeIdentifier      ::= Graphic String

-- A Record Element contains an attribute value derived from an attribute field from an object in the S-121
-- model. The record value may be of any of a set of predetermined types or may be a character string (which
-- could contain any information. The predefined types are: Direct Position, Coordinate, Date Time, Fraction,
-- Boolean.

RecordElement             ::= CHOICE {
    Graphic String,                                    -- Character string of the data value
    DirectPosition_FormattedCharacterString,          -- Direct Position data value
    Coordinate,                                       -- Single Latitude or Longitude data value
    DateTime,                                         -- Date or Time or Date and Time data value
    Fraction,                                         -- Fraction data value
    Boolean                                           -- Boolean data value
}

```

```
-- Structured or Formatted Character Strings -----
Oid_FormattedCharacterString
 ::= SEQUENCE {
     GraphicString           -- namespace
     DASH_Delimiter          -- dashDelimiter
     GraphicString           -- localId
 }
-- -----
-- A number of coordinate formats are permitted; however, it is recommended that only one coordinate format be
-- used for one type of coordinate in a dataset. Indicators of East / West and North / South may be either
-- E / W and N / S or + / -. When + and - are used the + is optional.

Coordinate
 ::= Choice { [1] DegreeMinuteSecondDecimalSecondsCardinal -- (N/S or E/W) DD MM SS.S
              [2] DegreeMinuteDecimalMinutesCardinal   -- (N/S or E/W) DD MM.M
              [3] DegreeDecimalDegreeCardinal          -- (N/S or E/W) DD.D
              [3] DegreeMinuteSecondDecimalSecondsSign -- (+ or -) DD MM SS.S
              [3] DegreeMinuteDecimalMinutesSign      -- (+ or -) DD MM.M
              [3] DegreeDecimalDegreeSign             -- (+ or -) DD.D
 }
DegreeMinuteSecondDecimalSecondsCardinal
 ::= SEQUENCE { CHOICE { Cardinal Direction
                           GraphicString           -- Cardinal Direction
                           SPACE                   -- {"N","S","E","W"}
                           GraphicString           -- Degrees
                           SPACE                  -- Space Delimiter
                           GraphicString           -- Minutes
                           SPACE                  -- Space Delimiter
                           GraphicString           -- Seconds
                           Decimal_Point_Delimiter OPTIONAL -- Decimal Point Delimiter
                           GraphicString           -- Decimal Seconds
 }
 }
DegreeMinuteDecimalMinutesCardinal
 ::= SEQUENCE { CHOICE { Cardinal Direction
                           GraphicString           -- Cardinal Direction
                           SPACE                   -- {"N","S","E","W"}
                           GraphicString           -- Degrees
                           SPACE                  -- Space Delimiter
                           GraphicString           -- Minutes
                           Decimal_Point_Delimiter OPTIONAL -- Decimal Point Delimiter
                           GraphicString           -- Decimal Minutes
 }
 }
DegreeDecimalDegreeCardinal
 ::= SEQUENCE { CHOICE { Cardinal Direction
                           GraphicString           -- Cardinal Direction
                           SPACE                   -- {"N","S","E","W"}
                           GraphicString           -- Degrees
                           Decimal_Point_Delimiter OPTIONAL -- Decimal Point Delimiter
                           GraphicString           -- Decimal Degrees
 }
 }
DegreeMinuteSecondDecimalSecondsSign
 ::= SEQUENCE { CHOICE { Signed Direction
                           GraphicString           -- Signed Direction "+","-"
                           SPACE                  -- Degrees
                           GraphicString           -- Space Delimiter
                           GraphicString           -- Minutes
                           SPACE                  -- Space Delimiter
                           GraphicString           -- Seconds
                           Decimal_Point_Delimiter OPTIONAL -- Decimal Point Delimiter
                           GraphicString           -- Decimal Seconds
 }
 }
DegreeMinuteDecimalMinutesSign
 ::= SEQUENCE { CHOICE { Signed Direction
                           GraphicString           -- Signed Direction "+","-"
                           SPACE                  -- Degrees
                           GraphicString           -- Space Delimiter
                           GraphicString           -- Minutes
                           Decimal_Point_Delimiter OPTIONAL {" "}
                           GraphicString           -- Decimal Point Delimiter
                           GraphicString           -- Decimal Minutes
 }
 }
DegreeDecimalDegreeSign
 ::= SEQUENCE { CHOICE { Signed Direction
                           GraphicString           -- Signed Direction "+","-"
                           Decimal_Point_Delimiter OPTIONAL -- Degrees
                           GraphicString           -- Decimal Point Delimiter
                           GraphicString           -- Decimal Degrees
 }
 }

Cardinal Direction
 ::= SEQUENCE { CHOICE {
     GraphicString {"N"}           -- Cardinal Direction
     GraphicString {"S"}           -- North indicator
     GraphicString {"E"}           -- South indicator
     GraphicString {"W"}           -- East indicator
     GraphicString {"W"}           -- West indicator
     SPACE OPTIONAL               -- The Cardinal Direction indicator
     -- of "N","S","E","W" may be left
     -- blank or be a Space. In this
   }
 }
```

```

-- case the value is assumed to
-- be "N" for Latitude and "E"
-- for Longitude.

Signed Direction
 ::= SEQUENCE { CHOICE {
    GraphicString {"+"}
    GraphicString {"-"}
    SPACE OPTIONAL
  }
}

-- Signed Direction
-- North or East indicator
-- South or West indicator
-- The Signed Direction indicator
-- of "+" or "-" may be left
-- blank or be a Space. In this
-- case the value is assumed to
-- be "+".

-- -----
-- DirectPosition_FormattedCharacterString
 ::= SEQUENCE { Coordinate -- Lat_position
                Slash_Delimiter
                Coordinate -- Long_position
  }

-- A structured character string represents date and time.
-- The format is YYYY:MM:DD:hh:mm:ss:timeZone
-- The sub fields are separated by the delimiter :
--   YYYY is the year e.g. 2017
--   MM is the month e.g. 05 (May)
--   DD is the day e.g. 17
--   hh is the hour e.g. 13 (in 24 hour clock)
--   mm is the minutes e.g. 15
--   ss is the seconds e.g. 00
--   timeZone is a shift from Greenwich Mean Time in hours
--   e.g. -5 for Eastern North America (EST).
--   The plus sign + is optional for east of Greenwich.
-- Therefore a date/time would be 2017:05:17:13:15:00:-5
-- Any unknown fields may be represented by consecutive delimiters e.g. :::
-- Least significant fields and delimiters may be omitted. E.g. a date may be 2018:05:12

DateTime
 ::= SEQUENCE { GraphicString OPTIONAL, -- centuryAndYear
                 COLON_Delimiter OPTIONAL, -- ColonDelimiter
                 GraphicString OPTIONAL, -- month
                 COLON_Delimiter OPTIONAL, -- ColonDelimiter
                 GraphicString OPTIONAL, -- day
                 COLON_Delimiter OPTIONAL, -- ColonDelimiter
                 GraphicString OPTIONAL, -- hour
                 COLON_Delimiter OPTIONAL, -- ColonDelimiter
                 GraphicString OPTIONAL, -- minute
                 COLON_Delimiter OPTIONAL, -- ColonDelimiter
                 GraphicString OPTIONAL, -- second
                 COLON_Delimiter OPTIONAL, -- ColonDelimiter
                 GraphicString OPTIONAL -- timeZone
  }

-- In an explicit text record oriented implementation a Fraction data type will be represented
-- as the text string "0.xxxx..." where xxxx... is the mantissa (fractional part of a number).

Fraction ::= GraphicString

-- -----
-- In an explicit text record oriented implementation a Boolean data type will be represented
-- as the text Yes or No.

Boolean ::= CHOICE {
  GraphicString {"Yes"},
  GraphicString {"No"}
}

-- ---- Terminators and Delimiters ---
TAB_Delimiter ::= Graphic String { IA5 Code 09}
COMMA_Delimiter ::= Graphic String { IA5 Code 44}
DASH_Delimiter ::= Graphic String { IA5 Code 45}
COLON_Delimiter ::= Graphic String { IA5 Code 58}
SLASH_Delimiter ::= Graphic String { IA5 Code 47}
Decimal_Point_Delimiter ::= Graphic String { IA5 Code 46}

```

```

SPACE ::= Graphic String { IA5 Code 32}
CR_Delimiter ::= Graphic String { IA5 Code 13}
LF_Delimiter ::= Graphic String { IA5 Code 10}
CRLF_Delimiter ::= SEQUENCE { CR_Delimiter, LF_Delimiter }
CRLF_CRLF_Delimiter ::= SEQUENCE { CRLF_Delimiter, CRLF_Delimiter }
CR_CR_Delimiter ::= SEQUENCE { CR_Delimiter, CR_Delimiter }
LF_LF_Delimiter ::= SEQUENCE { LF_Delimiter, LF_Delimiter }

-- The EndOfRecordMarker is the sequence of characters CR LF (or CR or LF).

EndOfRecordMarker
 ::= CHOICE {
    CRLF_Delimiter,           -- (IA5 Code 13)followed by (IA5 Code 10)
    CR_Delimiter,             -- (IA5 Code 13)
    LF_Delimiter              -- (IA5 Code 10)
}

The EndOfBlockMarker is the sequence of characters CR LF followed by CR LF (or CR followed by CR or LF followed by LF). This is equivalent to a blank record at the end of a block resulting in two EndOfRecordMarkers in a row.

EndOfBlockMarker
 ::= CHOICE {
    CRLF_CRLF_Delimiter,      -- (IA5 Codes 13,10,13,10)
    CR_CR_Delimiter,          -- (IA5 Code 13,13)
    LF_LF_Delimiter,          -- (IA5 Code 10,10)
}

The EndOfFile is the sequence of characters '==== End of File ===' followed by CR LF (or CR followed by CR or LF followed by LF). This is a block with a fixed content.

EndOfFile
 ::= SEQUENCE {
    EndOfFileMarker           -- A type of block that contains
                                -- the single record EndOfFileMarker
                                -- that contains an explicit string
                                -- identifying an end of file.
    EndOfBlockMarker           -- An EndOfFile block is terminated
                                -- by an EndOfBlockMarker like any other
                                -- block. That is the full end of file
                                -- sequence is '==== End of File ===CRLF'
}

EndOfFileMarker
 ::= Graphic String {"==== End of File ==="}

-- ----- Specific Block Types ---

-- The type of a block is identified in the BlockDescriptorRecord. The first two blocks are a MetadataBlock, -- and a GovernanceBlock. The MetadataBlock, and GovernanceBlock are identified by their location at the -- beginning of a data file. As such the content of the BlockDescriptorRecord for these blocks is optional. The -- BlockDescriptorRecord is required for each of these block types but it may be blank; that is, it may -- consist of an EndOfRecordMarker. Also the MetadataBlock, and GovernanceBlock are required, but they may be -- empty; that is they may consist of only an EndofBlockMarker. The contents of the metadata block is described -- the clauses 3.5.1 and 3.5.2. Structurally both the MetadataBlock, and GovernanceBlock are simply blocks.

MetadataBlock ::= Block
GovernanceBlock ::= Block
-----
```

Appendix F. Bibliography (Informative)

- [1] IHO S-57 IHO Transfer Standard for Digital Hydrographic Data, November (November 2000)
<<https://www.ihodata.org/standard/S-57Ed3.1/31Main.pdf>>
- [2] IHO S-100 IHO Universal Hydrographic Data Model (April 2017)
<https://www.ihodata.org/standard/S-100/S-100_Ed_3/S-100_Edition_3.0.0.pdf>
- [3] IHO S-101 IHO S-101 Electronic Navigational Chart Draft Product Specification
- [4] IHO S-121 Product Specification for Maritime Limits and Boundaries
- [5] ISO 8601 ISO 8601 Data elements and interchange formats—Information interchange—Representation of dates and times <<https://www.iso.org/iso-8601-date-and-time-format.html>>
- [6] ISO/IEC 8824-1:2015 Information technology -- Abstract Syntax Notation One (ASN.1)
<<https://www.iso.org/standard/68350.html>>
- [7] ISO 19108 ISO 19108:2002 Geographic information -- Temporal schema
<<https://www.iso.org/standard/26013.html>>
- [8] ISO 19115:2003 - Geographic information – Metadata <<https://www.iso.org/standard/26020.html>>
and ISO 19115-1:2014 Geographic information – Metadata Part 1: Fundamentals
<<https://www.iso.org/standard/53798.html>>
- [9] ISO/TS 19115-3:2016 Geographic information -- Metadata -- Part 3: XML schema implementation for fundamental concepts <<https://www.iso.org/standard/32579.html>>
- [10] ISO 19128:2005 Geographic information – Web map service interface (WMS)
<<https://www.iso.org/standard/32546.html>>
- [11] ISO 19136:2007 Geographic information -- Geography Markup Language (GML)
<<https://www.iso.org/standard/32554.html>>
- [12] ISO 19142:2010 - Geographic information – Web Feature Service <<https://www.iso.org/standard/42136.html>>
- [13] ISO 19152:2012 - Geographic information -- Land Administration Domain Model (LADM)
<<https://www.iso.org/standard/51206.html>>
- [14] OGC Catalogue Service for the Web (CSW) V3.0 OGC # 12-168r6, 12-176r7, 07-
<<http://www.opengeospatial.org/standards/cat>>
- [15] OGC Geography Markup Language (GML) Encoding Standard version 3.2.1 OGC # 07-036 with corrigendum 07-036r1 <https://portal.opengeospatial.org/files/?artifact_id=74183&version=2> together with GML - Extended schemas and encoding rules 10-129r1 <https://portal.opengeospatial.org/files/?artifact_id=46568>
- [16] OGC Web Feature Service (WFS) version 2.0.2 OGC # 09-025r1 with corrigendum 09-025r1
<<http://docs.opengeospatial.org/is/09-025r2/09-025r2.html>>
- [17] OGC OpenGIS Web Map Service (WMS) Implementation Specification OGC # 06-042
<http://portal.opengeospatial.org/files/?artifact_id=14416>

- [18] OGC Keyhole Markup Language (KML) 2.2 Document 12-147r2 <<http://www.opengeospatial.org/standards/kml>>
- [19] WC3 Extensible Markup Language (XML) <<https://www.w3.org/XML/>>
- [20] WC3 Extensible Stylesheet Language Transform (XSLT) <<https://www.w3.org/TR/xslt/all/>>
- [21] STANAG 7170 Ed: 3 Additional Military Layers (AML) – Digital Geospatial Data Products - AGeoP-19 Edition A <<http://nso.nato.int/nso/zPublic/stanags/CURRENT/7170EFed03.pdf>>
- [22] STANAG 4564 Warship Electronic Chart Display Information System (WECDIS)
<<http://nso.nato.int/nso/classDoc.htm>>
- [23] UN Convention on the Law of the Sea (UNCLOS)
<http://www.un.org/depts/los/convention_agreements/texts/unclos/unclos_e.pdf>
- [24] Esri Shape File, The shapefile (.shp) spatial data format, <<https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>>
- [25] Teledyne CARIS HIPS and SIPS data format <<http://www.teledynecaris.com/en/products/hips-and-sips/>>