INTERNATIONAL HYDROGRAPHIC ORGANIZATION

IHO TRANSFER STANDARD
for
DIGITAL HYDROGRAPHIC DATA

Edition 3.1 - November 2000

Special Publication No. 57

Published by the
International Hydrographic Bureau
MONACO

Page intentionally left blank

Important notice

All “Clarifications” in the latest Edition of the Maintenance Document must be taken into account before making use of this document.

INTERNATIONAL HYDROGRAPHIC ORGANIZATION

IHO TRANSFER STANDARD
for
DIGITAL HYDROGRAPHIC DATA

Publication S-57

Edition 3.1 - November 2000

PART 1	-	General Introduction
PART 2	-	Theoretical Data Model
PART 3	-	Data Structure
Appendix A	-	IHO Object Catalogue
Appendix B	-	Product Specifications

Published by the
International Hydrographic Bureau
4, Quai Antoine 1er
B.P 445 - MC 98011 MONACO Cedex
Principauté de Monaco
Telex: 479 164 MC - INHORG
Telefax: (+377) 93 10 81 40
E-mail: info@ihb.mc
Web http://www.iho.shom.fr

Page intentionally left blank

Important notice

All “Clarifications” in the latest edition of the Maintenance Document must be taken into account before making use of this document.

PREFACE

Edition 3.1 of the IHO Transfer Standard for Digital Hydrographic Data, Publication S-57, supersedes the previous Edition 3.0 dated November 1996. Edition 3.1 has been developed by the Transfer Standard Maintenance and Application Development Working Group (TSMAD) of the IHO Committee on Hydrographic Requirements for Information Systems (CHRIS).

The origin of Edition 3.1 can be traced back to November 1994 when the requirement for a detailed ENC Product Specification (PS) was clearly identified at the 6th CoE [1] Meeting. A workshop followed in February 1995 with participation of ECDIS manufacturers, hydrographic offices and regulatory authorities, with a view to agreeing on the ENC PS contents. At this workshop, it also became apparent that a new edition of S-57 was needed to satisfy both Hydrographic Offices and ECDIS manufacturers' requirements.

As a result, the DBWG [2] undertook the preparation of both Edition 3.0 of S-57 and the associated ENC PS. It met four times during 1995/1996 to complete the work. Changes from Version 2.0 included a binary implementation of the format, in addition to the existing ASCII one; a new cell structure concept; a revised updating mechanism based on unique object identifiers; and a more comprehensive data model.

The contents of the Standard were also significantly re-organized for Edition 3.0 so as to comprise a main section and two appendices. The main section is formed of three parts: a general introduction to the Standard, a theoretical data model and a description of the data structure. The two appendices contain the Object Catalogue of S-57 and Product Specifications (PS) for different applications. The PS for ENC is of particular interest to ECDIS manufacturers.

It was decided that Edition 3.0 should be frozen, i.e. remained unaltered, for four years from November 1996 onwards, in order to facilitate ENC data production by HOs and the development of ECDIS equipment by manufacturers.

Based on practical experience, some Hydrogaphic Offices identified a limited number of attribute values that they required for ENC purposes but which were not contained in Edition 3.0. It was therefore agreed that a limited new edition of S-57, Edition 3.1, would be published at the end of the four-year period, which would incorporate only these additional attribute values.

S-57 Edition 3.1 was officially made available in November 2000. A familiarization version was however distributed in November 1999 to allow data producers and equipment manufacturers time to familiarize themselves with the new content. Edition 3.1 will remain frozen until at least November 2002.

1 *Former Committee on ECDIS, now redefined as "CHRIS".*
2 *Data Base Working Group, now redefined as "TSMAD".*

All components of the Standard will remain unchanged for the duration of the freeze period, with the exeption of the “Use of the Object Catalogue for ENC”, “IHO Codes for Producing Agencies”, “Recommended ENC Validation Checks” and “INT 1 to S-57 Cross Reference” documents. Subsequent editions of which will be published as required in the course of the freeze period. Any “Corrections” or “Extensions” required for future editions of S-57, and any “Clarifications” to the existing frozen text, will be recorded in the “cumulative Maintenance Document”. (Available from the IHO web site: http://www.iho.shom.fr). The MD document is usually updated after meetings of the TSMAD Working Group.

RAdm Neil GUY, IHB Director
Chairman, IHO CHRIS
November 2000

CONTENTS

Appendixes

Page intentionally left blank

S-57 Part 1

General Introduction

Edition 3.1

Page intentionally left blank

Important notice

All "Clarifications" in the latest edition of the Maintenance Document must be taken into account before making use of this document.

1. Scope

The publication “S-57—IHO Transfer Standard for Digital Hydrographic Data” describes the standard to be used for the exchange of digital hydrographic data between national hydrographic offices and for its distribution to manufacturers, mariners and other data users. For example, this standard is intended to be used for the supply of data for ECDIS. This transfer and distribution has to take place in such a way that none of the meaning of the data is lost.

The Standard was prepared by the International Hydrographic Organization's (IHO) Committee on Hydrographic Requirements for Information Systems (CHRIS).

The Standard was adopted as the official IHO standard, by the XIVth International Hydrographic Conference, Monaco, 4-15 May 1992.

2. Structure of the standard

The contents of this standard are organized as follows:

Part 1 provides a general introduction including a list of references and definitions of terms used in the rest of the Standard.

Part 2 describes the theoretical data model on which the entire Standard is based.

Part 3 defines the data structure or format that is used to implement the data model and defines the general rules for encoding data into that format.

The Standard also has two appendixes:

Appendix A is the Object Catalogue. This provides the official, IHO approved data schema that can be used within an exchange set to describe entities in the real world.

Appendix B contains the IHO approved Product Specifications. These are additional sets of rules applicable to specific applications.

3. ISO/IEC 8211 encapsulation

S-57 uses the international standard ISO/IEC 8211 (*“Specification for a data descriptive file for information interchange”*) as a means of encapsulating data. The ISO/IEC 8211 standard provides a file based mechanism for the transfer of data from one computer system to another, independent of make. In addition, it is independent of the medium used to establish such a transfer. It permits the transfer of data and also the description of how such data is organized.

4. References

This Standard is based upon the specifications contained in the standards listed below. All standards are subject to revision, therefore, parties to agreements based on this Standard are encouraged to use the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid international standards.

ISO/IEC 646,	*“Information technology - ISO 7-bit coded character set for information interchange”*
ISO 2022,	*“Information processing - ISO 7-bit and 8-bit coded character sets - Code extension techniques”*
ISO 2375,	*“Data processing - Procedure for registration of escape sequences”*
ISO 6093,	*“Information processing - Representation of numerical values in character strings for information interchange”*
ISO 6937,	*“Information processing - Coded character sets for text communication”*
ISO/IEC 8211,	*“Information processing - Specification for a data descriptive file for information interchange”*
ISO 8859,	*“Information processing - 8-bit single-byte coded graphic character sets Part 1: Latin alphabet No.1"*
ISO/IEC 10646,	*“Information technology - Universal Multiple-Octet Coded Character Set (UCS) Part 1: Architecture and Basic Multilangual Plane”*

5. Definitions

The following are definitions of various terms as used in S-57.

Application profile A defined subset of the S-57 data structure.

Applier The entity (software) controlling the application of the *update information*.

Attribute A characteristic of an *object*. It is implemented by a defined *attribute label/code*, acronym, definition and applicable values (see Appendix A, IHO Object Catalogue). In the data structure, the attribute is identified by its label/code. The acronym is only used as a quick reference in related documents and *product specifications*. Attributes are either qualitative or quantitative.

Attribute label/code A fixed length numeric label or a 2-byte unsigned integer code of an *attribute* as defined in Appendix A, the IHO Object Catalogue.

Base data S-57 conforming data at the data producer’s site that does not contain any *update records*. Once this data is exchanged it becomes *target data* at the *applier’s* site.

Blank ASCII space (2/0).

Chain-node *) Data structure in which the geometry is described in terms of *edges, isolated nodes* and *connected nodes.* Edges and connected nodes are topologically linked. *Nodes* are explicitely coded in the data structure.

*) Previously known in S-57 2.0 as chain-explicit-node.

Connected node A *node* referred to as a beginning and/or end node by one or more *edge*. Connected nodes are defined only in the *chain-node*, *planar graph* and *full topology* data structures.

Data set A logical grouping of S-57 data to which the S-57 data set descriptive records apply. The data set descriptive records contain meta data. The use of data set descriptive records is product specific and is, therefore, defined by a *product specification*. If the data set descriptive records are repeated for each file in an *exchange set*, an instance of a file containing the data set descriptive records is called a data set. If the data set descriptive records are encoded generally for the whole exchange set, the exchange set is referred to as a data set.

Domain The set of all permissible values for an *attribute*.

Edge A one-dimensional *spatial object*, located by two or more coordinate pairs (or two *connected nodes*) and optional interpolation parameters. If the parameters are missing, the interpolation is defaulted to straight line segments between the coordinate pairs. In the *chain-node*, *planar graph* and *full topology* data structures, an edge must reference a connected node at both ends and must not reference any other *nodes*.

Encapsulation The identification of *fields* and *records* and the grouping of fields and records and the data syntax rules used.

Enumerated A *domain* defined by an arbitrary value; each value is explicitly listed and described.

Exchange set The set of *files* representing a complete, single purpose (i.e. product specific) data transfer. For example, the "ENC product specification" defines an exchange set which contains one Catalogue file and at least one Data Set file.

Face A two dimensional *spatial object*. A face is a continuous area defined by a loop of one or more *edges* which bound it. A face may contain interior holes, defined by closing loops of edges. These interior boundaries must be within the outer boundary. No boundary may cross itself or touch itself other than at the beginning/end *node*. None of the boundaries may touch or cross any other boundary. Faces are defined only in the *full topology* data structure.

Feature object An *object* which contains the non-locational information about real world entities. Feature objects are defined in Appendix A, IHO Object Catalogue.

Feature record A feature record is the implemented term used in the S-57 data structure for a *feature object* (i.e. a feature object as defined in the data model is encoded as a feature record in the data structure). There are four types of feature records: geo, meta, collection and cartographic.

Field A named collection of labeled *subfield(s)*. For example, IHO attribute label/code and IHO Attribute Value are collected into a field named Feature Record Attribute.

File An identified set of S-57 *records* collected together for a specific purpose. The file content and structure must be defined by a *product specification*.

Foreign key The *key* or the *object identifier* of one *record* and held by another record as a reference. In the case of a reference between *feature records*, the foreign key is always the object identifier of the referenced feature record. In all other cases the foreign key is the key of the referenced record. The foreign key is held in the *pointer* field.

Full topology A 2-dimensional data structure in which the geometry is described in terms of *nodes*, *edges* and *faces* which are all *topologically* linked. A *planar graph* with faces.

Geometric primitive One of the three basic geometric units of representation: point, line and area.

Isolated node An isolated zero-dimensional *spatial object* that represents the geometric location of a point feature. An isolated node is never used as a beginning or end *node*.

ISO/IEC 8211 Record An ISO/IEC 8211 implementation of a S-57 *record* and which comprises one or more *fields*.

Key The identification of a S-57 *record* which is the concatenation of the "Record name" and "Record Identification Number". Within the context of this Standard the key is referred to as the "Name" [NAME].

Label An ISO/IEC 8211 implementation concept used to identify the *subfield*.

Note - The data descriptive part of an ISO/IEC 8211 file may include a cartesian label which when expanded gives rise to labels for each data item of an array.

Matrix An array of regularly spaced locations.

Node A zero-dimensional *spatial object*, located by a coordinate pair. A node is either *isolated* or *connected*.

Object An identifiable set of information.

Object identifier The identification of a S-57 *feature object*. The object identifier is the concatenation of the "Producing Agency", "Feature Identification Number" and "Feature Identification Subdivision" subfields. Within the context of this Standard the object identifier is referred to as the "Long Name" [LNAM].

Object Class A generic description of *objects* which have the same characteristics.

Planar graph A 2-dimensional data structure in which the geometry is described in terms of *nodes* and *edges* which are *topologically* linked. A special case of a *chain-node* data structure in which edges must not cross. *Connected nodes* are formed at all points where edges meet.

Pointer The S-57 data structure implementation of a *relationship*. A pointer establishes a link between two *records* (e.g. a *feature record* can be related to a *spatial record* by means of a pointer). The pointer contains the *foreign key*. Within S-57 all pointers are one-way as indicated in the data model.

Pre-order Traversal Sequence Representation of the order in which information, in a tree structure diagram, must be interpreted. The sequence is extremely important and inviolate as there is no other explicit method of specifying the interfield (parent/child) relationships within the ISO/IEC 8211 data records.

Product specification A defined subset of the entire specification combined with rules, tailored to the intended usage of the transfer data.

Raster A regular array with information pertaining to each element (pixel) or group of elements.

Record A S-57 construct which is comprised of one or more tagged S-57 *fields* and identified by a *key*.

Relationship A logical link between two elements from the data model which may be spatial (e.g. topological relationship) and/or non-spatial. In general a relationship is implemented in the data structure as a *pointer*.

Spaghetti A data structure in which all lines and points are unrelated to each other (i.e. no topological relationships exist in the data structure).

Spatial object An *object* which contains locational information about real world entities.

Spatial record A spatial record is the implemented term used in the S-57 data structure for a *spatial object* (i.e. a spatial object as defined in the data model is encoded as a spatial record in the data structure). There are three types of spatial records: *vector*, *raster* and *matrix*.

Subfield A subfield is a component of a *field*. It is a contiguous string of bytes whose position, length and data type are described in the field data description. It is the smallest unit of information which can be described by this standard.

Note - Certain stylized subfields, such as date (YYYYMMDD), must be further resolved by an application.

Table A 2-D array with a fixed number of columns and indefinitely repeating, unlabeled rows all having the same format. The columns are labeled and the number of columns may be one. Alternatively, both the columns and rows are labeled and describe an array of fixed dimension.

Tag An ISO/IEC 8211 implementation concept used to identify each instance of a *field*.

Target data Data on which an *update operation* is performed by the *applier*.

Target record A *feature* or *spatial record* on which an *update operation* is performed by the *applier*.

Topology A branch of mathematics that investigates the properties of a geometric configuration that are unaltered if the configuration is subjected to a transformation continuous in both directions.

Update information The data which are needed to update the *target data* automatically. Update information comprises one or more *update records*.

Update record An update record is a generic term for *feature* or *spatial records* containing update instructions.

Update mechanism The defined sequence of *update operations* necessary to update the *target data* by applying the *update information* to the content of the *target data* so that no operator interaction is involved.

Update operation The application of a single *update record*.

Update process The controlled performance of the *update mechanism*.

Vector Spatial information whose data model is based on graph theory.

Volume An exchangeable physical unit of storage media (e.g. a reel of magnetic tape). A volume may contain part of a *file*, a complete file or more than one file.

6. Conformance

An exchange set conforms to this standard when all of its contents conform to the specifications of this standard. Requirements for specific user applications must be defined by a product specification. When included in Appendix B, these product specifications form an integral part of this standard. An exchange set based on any of these product specifications, therefore, conforms to this standard.
Any statement of conformance must specify the edition number of this standard and the product specification on which it is based.

7. Maintenance

Changes to this Standard are coordinated by the "Transfer Standard Maintenance Working Group" (TSMWG) of the IHO. National hydrographic offices which wish changes to be made to the standard, either to correct errors which they have identified or to enhance its applicability, must address their comments to the International Hydrographic Bureau. Other users of the Standard, for example equipment manufacturers, must address their comments to their national hydrographic office. (Addresses of IHO Member States' hydrographic offices can be found in IHO Yearbook, publication P-05).

The International Hydrographic Bureau maintains the Standard by means of the following three documents:

Clarifications Document. This contains improvements to the wording of the Standard. These are editorial amendments which do not result in any substantive change to the Standard. The clarifications document is distributed with the Standard and its contents are also available on the IHB's Bulletin Board System (Tel: 377 93 10 81 27-28) and on the IHO Web site (http://www.iho.shom.fr).

Corrections Document. This contains changes to the Standard to correct factual errors and make necessary amendments to the Standard. The corrections document is distributed with the Standard and its contents are also available on the IHB's Bulletin Board System (Tel: 377 93 10 81 27-28) and on the IHO Web site (http://www.iho.shom.fr).

Extensions Document. This contains extensions, or other significant changes to the Standard, which have been agreed by the appropriate IHO committee or working group and will be included in the next edition of the standard. This is a working document which is only available on demand.

These documents, and the associated maintenance mechanism, do not apply to the product specifications contained in Appendix B of this Standard. The maintenance procedure for a particular product specification is described in that specification.

S-57 Part 2

Theoretical data model

Edition 3.1

Page intentionally left blank

Important notice

All "Clarifications" in the latest edition of the Maintenance Document must be taken into account before making use of this document.

1. Model introduction

This standard is designed to permit the transfer of data describing the real world. The real word is far too complex for a complete description to be practical, therefore a simplified, highly-specific, view of the real world must be used. This is achieved by modeling the reality.

This standard is specifically concerned with those entities in the real world that are of relevance to hydrography. This hydrographic regime is considered to be geo-spatial. As a result, the model defines real world entities as a combination of descriptive and spatial characteristics. Within the model these sets of characteristics are defined in terms of feature objects and spatial objects.

An object is defined as an identifiable set of information. An object may have attributes and may be related to other objects.

Feature objects contain descriptive attributes and do not contain any geometry (i.e. information about the shape and position of a real world entity). Spatial objects may have descriptive attributes and must contain geometry.

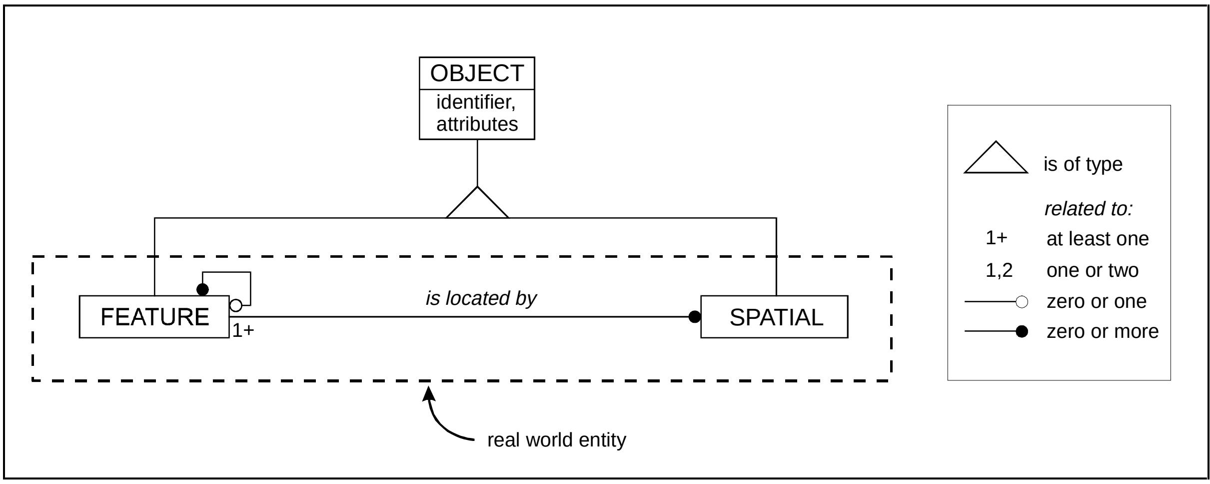

*figure 1.**1***

A feature object is located by a relationship to one or more spatial objects. A feature object may exist without referencing a spatial object, but each spatial object must be referenced by a feature object.

2. Model implementation

The following diagram represents the overall model used by this standard. It is further explained by clauses 2.1 to 2.2.

2.1 Feature objects

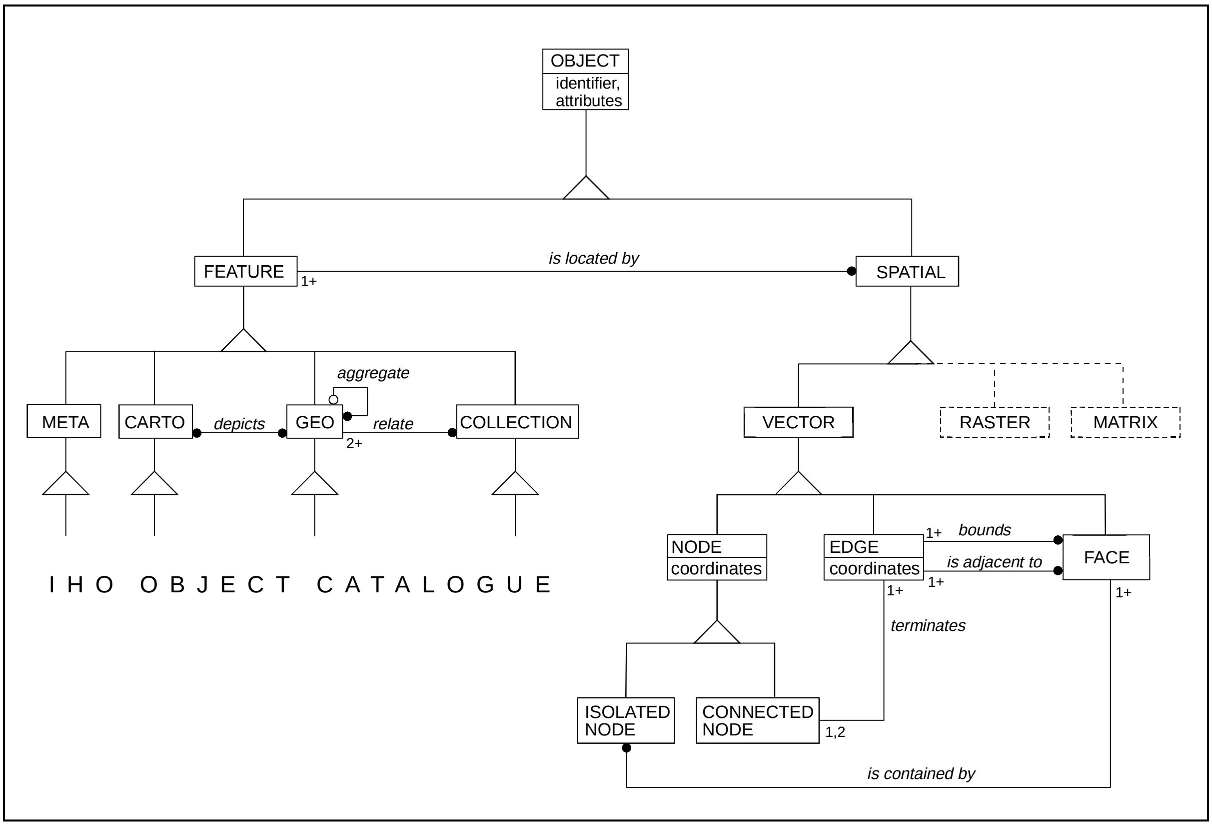

*figure 2.**1***

To facilitate the efficient exchange of the non-locational description of the real world entities, this model defines four types of feature objects:

Meta	Feature object which contains information about other objects.
Cartographic	Feature object which contains information about the cartographic representation (including text) of real world entities.
Geo	Feature object which carries the descriptive characteristics of a real world entity.
Collection	Feature object which describes the relationship between other objects.

The sub-types of these feature objects are defined in Appendix A (IHO Object Catalogue).

2.2 Spatial objects

There are several ways of representing the spatial characteristics of a real world entity. Within this model these representations are limited to vector, raster and matrix. Therefore, spatial objects can be of type vector, raster or matrix.

2.2.1 Vector model

In order to further simplify the model a two dimensional planar view of reality is taken. Therefore, spatial objects of type vector can have zero, one or two dimensions which are implemented as nodes, edges and faces respectively. The third dimension is expressed as an attribute of an object.

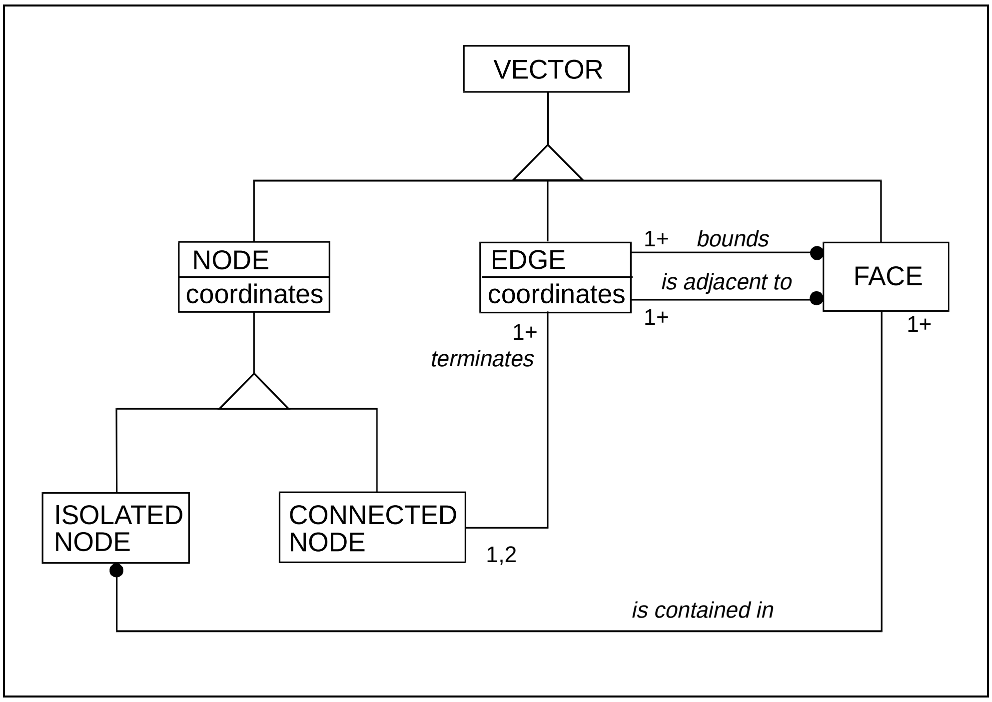

figure 2.2

In above diagram (figure 2.2) the following relationships are defined:

Isolated node .... is contained in .... Face
Face........................... contains.............................. Isolated node
Edge .... bounds .... Face
Face........................... is bounded by...................... Edge
Connected node .... terminates .... Edge
Edge .... is terminated by.................. Connected node
Edge .... is adjacent to....................... Face

These relationships can be used to describe four levels of topology:

- Cartographic spaghetti
- Chain-node
- Planar graph
- Full topology

These levels are described in clauses 2.2.1.1 to 2.2.1.4. A diagram illustrating the topological relationships is given for each level of topology. In these diagrams the relationships “point representation”, “line representation” and “area representation” are specializations of the relationship “is located by” in the overall data model diagram (figure 2.1).

2.2.1.1 Cartographic spaghetti

A set of isolated nodes and edges. Edges do not reference nodes. Feature objects must not share spatial objects. Point representations are coded as isolated nodes. Line representations are coded as connected series of edges. Area representations are coded as closing loops of edges. If logical consistency is required, coincident edges must contain identical geometry. The cartographic spaghetti model is illustrated in figure 2.3.

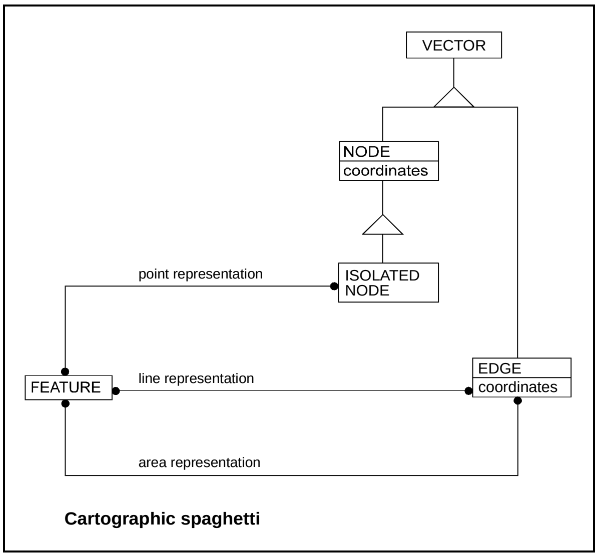

figure 2.3

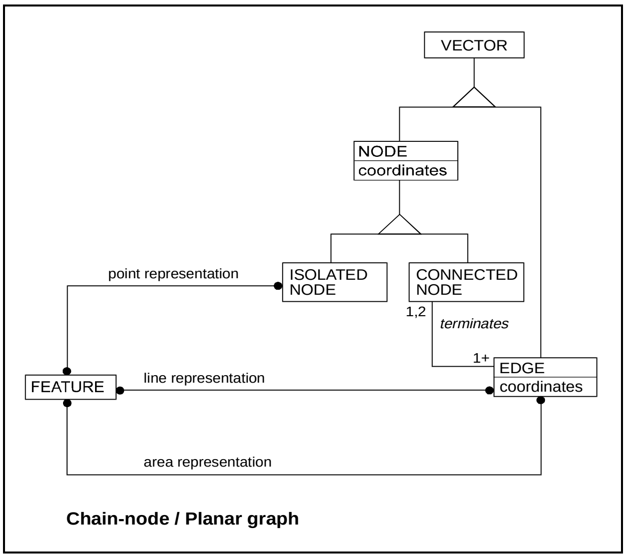

figure 2.4

2.2.1.2 Chain-node

A set of nodes and edges. Each edge must reference a connected node as its beginning and end (they may be the same node). The geometry of the referenced node is not part of the edge (see clause 4.7.2). Vector objects may be shared. Point representations are coded as nodes (isolated or connected). Line representations are coded as series of edges and connected nodes. Area representations are coded as closing loops of edges starting and ending at a common connected node. Duplication of coincident linear geometry is prohibited. The chain-node model is illustrated in figure 2.4.

2.2.1.3 Planar graph

A set of nodes and edges. A chain-node set where edges must not cross and may touch only at the connected nodes. Vector objects may be shared, with the restriction that touching edges always share connected nodes and adjacent areas always share the edges forming their common boundary. Duplication of coincident geometry is prohibited. The planar graph model is illustrated in figure 2.4.

2.2.1.4 Full topology

A set of nodes, edges and faces. A planar graph with defined faces. The universe is partitioned into a set of mutually exclusive and collectively exhaustive faces. Isolated nodes may reference their containing faces and edges must reference the faces to their right and left. Point representations are coded as nodes (isolated or connected). Line representations are coded as series of edges and connected nodes. Area representations are coded as faces. Duplication of coincident geometry is prohibited. The full topology data model is illustrated in figure 2.5.

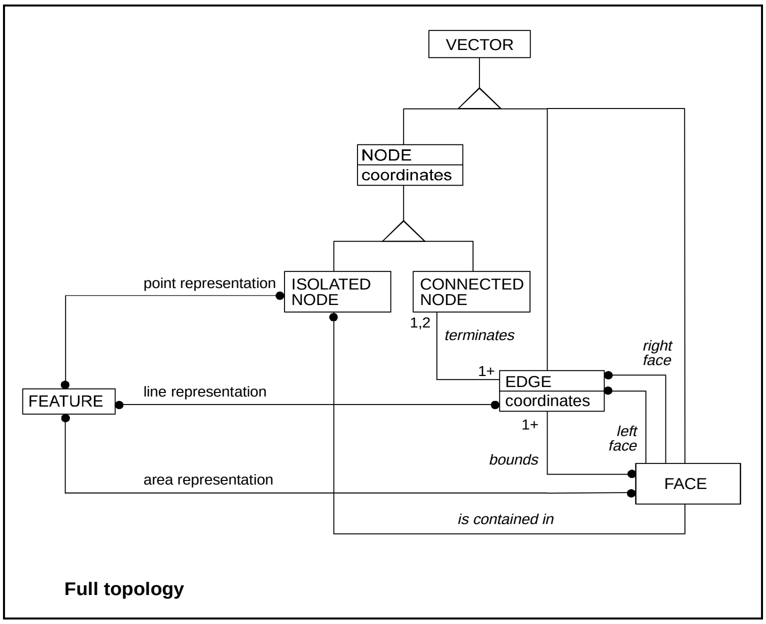

figure 2.5

2.2.2 Raster model

To be defined.

2.2.3 Matrix model

To be defined.

3. Presentation

The model described in this part of the Standard does not contain any rules for the presentation or display of information. It provides only the means for the factual description of the real world. The presentation of this information may vary to suit a particular use (e.g. it may be presented either graphically, using symbols, or in a textual form). Therefore, the presentation of information is considered to be independent of its storage. Different applications must provide their own specific “presentation models”. A presentation model defines, via a set of presentation rules, the way in which real world information must be displayed for a specified application. The concept of keeping information storage independent of presentation provides for greater versatility and flexibility. It allows the same data to be used for many purposes without requiring any change to its structure or content. If the presentation style or medium changes, only the presentation model has to be changed.

Therefore, the model described can be linked to many different presentation models. For example, ECDIS and paper charts present the same basic data in different ways via different presentation models.

Page intentionally left blank

S-57 Part 3

Data Structure

Edition 3.1

Page intentionally left blank

Important notice

All "Clarifications" in the latest edition of the Maintenance Document must be taken into account before making use of this document.

1. Structure introduction

This part of the standard specifies how the Theoretical Data Model (see part 2) is translated into the S-57 data structure. This translation implies a linkage between logical constructs taken from the model and physical constructs used in the structure. That linkage is explained in this chapter.
Although the translation from model to structure is independant of use, each application or product specifies its own rules for implementing the data structure. This set of rules is called a product specification. The concept of product specifications is introduced in this chapter (see clause 1.4).

1.1 Model to structure translation

In order to transfer information about the real world, a layered approach is used.

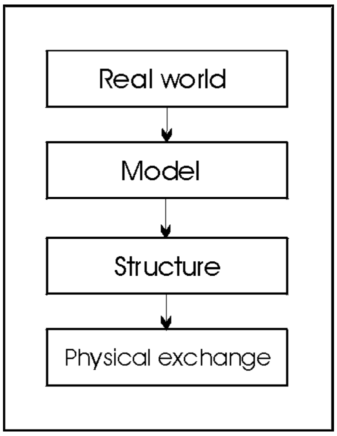

figure 1.1

First the real world is simplified by modeling reality. This step is described in part 2 of this standard (Theoretical Data Model). The resulting model is translated into named constructs (e.g. records and fields). Part of that translation is the definition of rules and constraints for both the constructs and their content. The translation results in a data structure.
The data structure itself can not be translated directly from one computer system to an other. To allow for that, the structure must be encapsulated in a physical transfer standard. S-57 uses ISO/IEC 8211 as its encapsulation.

This chapter deals with the implementation of the data model (i.e. the translation from model to structure). The relationship between the model and structure constructs is shown below.

Model	**Structure**
Feature object	Feature record
Meta feature object	Meta feature record
Cartographic feature object	Cartographic feature record
Geo feature object	Geo feature record
Collection feature object	Collection feature record
Spatial object	Spatial record
Vector object	Vector record
Isolated node object	Isolated node vector record
Connected node object	Connected node vector record
Edge object	Edge vector record

Face object	Face vector record or (loop of) edge vector record(s)
Raster object	Raster record
Matrix object	Matrix record
Attributes	Feature or spatial attribute field
Relationship between feature objects	Collection feature record or pointer field
Relationship between feature and spatial objects	Pointer field

Usually, more than one object is involved in an exchange. Therefore, since an object is structured into a record, an exchange is comprised of more than one record. To facilitate this, records are grouped into files. The set of information which is finally exchanged is called an exchange set.
The way in which records are grouped into files and files are grouped into exchange sets is considered to be application specific (see clause 1.4). However, the following general rules apply:

- an exchange set is formed of one or more files;
- a file is formed of one or more records;
- a record is formed of one or more fields;
- a field is formed of one or more subfields.

The hierarchy is shown below

```
 Exchange set
     |                     Product specification
     |---<R>--File(s)
               |
.................................................................................
               |---<R>--Record(s)
                         |
                         |---<R>--Field(s)
                                   |
                                   |---<R>--Subfields(s)
     S-57 data structure


<R> construct is allowed to repeat
```

figure 1.2

A dotted line is shown in figure 1.2. The constructs below the dotted line are part of the S-57 data structure and are specified in detail in chapter 7. The constructs above the line are specific to an application or product and are, therefore, defined by the relevant product specification (see clause 1.4 and Appendix B).

The lowest level construct, the subfield, must only contain one elementary data item, for example, one attribute value. Formatted subfields, such as date subfields, must be further resolved by an application program. In this specification, such subfields are not divisible.

The S-57 data structure has both ASCII and binary implementations. The implementation to be used for a specific exchange is defined by the relevant product specification.

1.2 Records

1.2.1 General

This standard defines a set of records from which an exchange set can be built. These records fall into five categories:

- Data Set Descriptive (meta)
- Catalogue
- Data Dictionary
- Feature
- Spatial

A detailed description of the contents of these records is given in chapter 7. Clauses 1.2.2 to 1.2.6 give a short description. The exact use of records, fields and subfields may be further defined by a product specification. For example, a product specification may prohibit the use of certain records, fields and subfields.

1.2.2 Data set descriptive records

The data set descriptive (meta) records contain the following information:

- Information required to identify the general purpose and nature of the exchanged information. (For example, the product specification employed is defined in this part).
  [Data Set General Information record (see clause 7.3.1)]

- Information about the coordinate system, the projection, the horizontal and vertical datums used, the source scale and the units of height and depth measurement
  [Data Set Geographic Reference record (see clause 7.3.2)]

- Information about the origins of the data set.
  [Data Set History record (see clause 7.3.3)]

- Information describing the accuracy of the locational data in the spatial records.
  [Data Set Accuracy record (see clause 7.3.4)]

Information held in these records defines the default for the data set. The default can be overruled by more detailed information at the feature record level using either meta feature records or attributes of an object (see clause 1.3).

1.2.3 Catalogue records

The catalogue records contain the following information:

- Information required to allow the decoder to locate and reference files within the entire exchange set. This part can be compared with a table of contents.
  [Catalogue Directory record (see clause 7.4.1)]

- Information about special relationships between individual records within the exchange set. [Catalogue Cross Reference record (see clause 7.4.2)]

S-57 has both binary and ASCII implementations. The implementation to be used is specified in the "Catalogue Directory" record. To ensure correct interpretation, the “Catalogue Directory” record is always encoded using the ASCII implementation.

1.2.4 Data dictionary records

The data dictionary records contain the description of objects, attribute and attribute values used in an exchange set. It is not necessary to utilize these records in an exchange set if the IHO Object Catalogue (Appendix A) is used exclusively. However, if any non-IHO Object Catalogue objects, attributes or attribute values are used in an exchange set, they must be described in these records (see clause 2.3). The Data Dictionary records contain the following information:

- Information defining the object classes and attributes.
  [Data Dictionary Definition record (see clause 7.5.1)]

- Information about the attribute value domains.
  [Data Dictionary Domain record (see clause 7.5.2)]

- Information identifying which attributes are valid for an object class.
  [Data Dictionary Schema record (see clause 7.5.3)]

1.2.5 Feature records

The feature records contain the non-locational real-world data. They may be of type meta, cartographic, geo or collection.

Feature records include the following information:

- The object information describing the real world, including relationships and update instructions.
  [Feature record (see chapter 4 and clause 7.6)]

1.2.6 Spatial records

The spatial records contain the locational data. They may be of type vector, raster or matrix. An exchange set may contain a mixture of the different spatial record types.

1.2.6.1 Vector records

Vector records include the following information:

- The coordinate geometry related to the feature records, including spatial attributes, topological relationships and update instructions. Vector records may be of type node, edge or face.
  [Vector record (see clause 5.1 and clause 7.7.1)]

1.2.6.2 Raster records

To be defined.

1.2.6.3 Matrix records

To be defined.

1.3 Meta data

Meta data can be provided at three levels within an exchange set. These levels are defined below. Also defined below are the rules that apply when meta data provided at different levels conflicts:

a) Meta information which is defined in the data set descriptive records (see clause 1.2.2) provides the default for the data set in which those records are contained. Whether data set descriptive records are defined at the file or the exchange set level depends on the product specification used.

b) Information which is defined by meta objects overrides the default defined by the data set descriptive records. Meta objects are defined in the IHO Object Catalogue (Appendix A) and are encoded as feature objects.

c) Information which is defined by attributes of individual objects overrides that defined by meta objects and, therefore, that derived from the data set descriptive records.

1.4 Rules for specific applications

1.4.1 Product specifications

This Standard is intended to support all hydrographic applications. However, different applications have different requirements for data transfer. To establish effective data transfer, additional rules may be defined for specific applications or products. These sets of rules are called product specifications.

All product specifications which have been officially approved by the IHO are included in Appendix B and form an integral part of this standard. The product specification used for a given exchange must be specified in either the PRSP subfield or the PSDN subfield of the DSID field (see clause 7.3.1.1).

All data within an exchange set must be based on the same product specification.

1.4.2 Application profiles

A basic part of a product specification is the definition of a subset of the data structure defined in chapter 7. The mechanism used to define this subset is called an "Application Profile". It is strongly recommended that this mechanism is used in all product specifications. It ensures an easy and efficient decoding of data. For example, a product specification might define two possible application profiles, one for the initial supply data and one for updates.

At present, three application profiles have been defined:

EN {1} digital data comprising the basic Electronic Navigational Charts (ENC), used to populate the SENC

ER {2} digital data revising the SENC

DD {3} digital data comprising a machine readable version of the IHO Object Catalogue

Note: The abbreviations (ASCII, e.g. **EN**) and the numbers between parentheses (binary, e.g. {1}) are possible values for the PROF subfield of the DSID field (see clause 7.3.1.1).

Other application profiles may be established in the future as additional categories of data transfer are identified. Values for new or non-conforming application profiles can be defined by an appropriate product specification.

The encoder of a conforming exchange set must pay particular attention to its content to ensure that it is consistent with the stated application profile.
Records, fields and subfields not marked in an application profile are optional (unless otherwise specified in the product specification). However, when less than the full field contents are required, any omitted subfields must be accounted for in the data records (see clause 2.1).

2. General coding conventions

The conventions listed in this chapter are a set of rules according to which conforming exchange sets must be encoded. These conventions must always be followed by an encoder.

2.1 Omitted constructs

For a given application profile, the inclusion of certain records, fields and subfields in an exchange set may be optional (see Appendix B – Application Profiles). The encoder may, therefore, choose to omit these constructs from the exchange set. The omission of any optional constructs from an exchange set must be interpreted as follows:

Missing record: The content of the omitted record is not meaningful for the intended use as determined by the producer.
No specific measures are needed to account for the missing record in the exchange set.

Missing field: The content of the omitted field is not meaningful for the intended use as determined by the producer.
No specific measures are needed to account for the missing field in the exchange set.

Missing subfield value: The omitted subfield value is either not known or is not meaningful for the intended purpose as determined by the producer. The exact interpretation must be defined by the relevant product specification.
In S-57, the data description used for each field is independent of the application profile employed (i.e. subfields can not be omitted). Therefore, each missing subfield value must be accounted for in the exchange set. The way this is done depends on the type of subfield (see table 2.1).

Subfield type	ASCII data format (see 7.2.2.1)	Binary data format (see 7.2.2.1)
Fixed length	The space normally occupied by the subfield must be filled with blanks	The binary value with all bits set to 1 must be used.
Variable length	Only the subfield delimiter must be encoded	Not applicable

table 2.1

Note : Fields and subfields values marked as mandatory for a given application profile (as documented in Appendix B – Product Specifications) must not be omitted.

2.2 Record identifier

The record identifier is the basic identifier in the data structure and is used to maintain topological relationships and to address information contained in update messages. Every record in an exchange set based on S-57 must have an identifier. For this purpose the “Record Name” [RCNM] and the “Record Identification Number” [RCID] subfields are used. These subfields must be encoded in the first field of every record.

The concatenation of these subfields forms the identification (key) of a record and is known as the “Name” [NAME]. This subfield is used as a foreign pointer. In the binary implementation, NAME is a bit string with a length of 40 bits (i.e. the “binary” concatenation of the RCNM and RCID subfields). Retrieval of the individual subfield values from the bit string must be resolved by the decoder.

Records are grouped into files (see clause 1.1). The concatenation of the RCNM and RCID subfields must, at least, be unique within the file in which the record is contained.

The NAME subfield is not used for relationships between feature objects. For this purpose the "Feature Object Identifier" is used (see clause 4.3).

2.2.1 Record name [RCNM]

The record names used in this standard are listed in table 2.2.

Record name		ASCII value	Binary value
Data Set General Information		DS	{10}
Data Set Geographic Reference		DP	{20}
Data Set History		DH	{30}
Data Set Accuracy		DA	{40}
Catalogue Directory		CD	*)
Catalogue Cross Reference		CR	{60}
Data Dictionary Definition		ID	{70}
Data Dictionary Domain		IO	{80}
Data Dictionary Schema		IS	{90}
Feature		FE	{100}
Vector	Isolated node	VI	{110}
	Connected node	VC	{120}
	Edge	VE	{130}
	Face	VF	{140}

table 2.2

*) Only ASCII data types are allowed in the Catalogue Directory record.

2.2.2 Record identification number [RCID]

The "Record Identification Number" ranges from 1 to 2^{32}-2. The only constraint on usage of this subfield is that the concatenation of the RCID and RCNM subfields must be unique within the file in which the record is contained.

2.3 Use of non-Object Catalogue codes and values

Every effort must be made to encode the features of an exchange set using the object classes, attributes, and attribute values identified in the IHO Object Catalogue (see Appendix A). However, where a feature cannot sensibly be described using values from the Object Catalogue, the encoder may define new object classes, attributes and attribute values, provided the following conditions are adhered to:

- none of the definitions associated with the object classes or attributes in the IHO Object Catalogue is satisfactory for use;
- the label/code and acronym used are clearly distinguishable from the labels/codes and acronyms reserved by the IHO Object Catalogue for both objects and attributes;
- All non-standard object and attributes labels/codes must be in the range 16388 to 65534;
- All non-standard object and attribute acronyms must be lowercase ISO/IEC 646 IRV characters;
- the non-standard usage is fully documented within the Data Dictionary Definition, Domain and Schema records (see clause 7.5).

All labels/codes and acronyms currently defined in the Object Catalogue are reserved by this Standard. The labels/codes 8193 to 16387 are reserved for further standardization.

Extensions to usage are permitted, however, redefinition or contradictory use of reserved labels/codes or acronyms is *not* permitted.

The use of non-Object Catalogue names and values may be further constrained by a product specification.

In all cases where this standard refers to the IHO Object catalogue, other data schema (catalogues) may be used instead, provided that the rules of this clause are adhered to.

2.4 Use of character sets

The default character set which must be used for all non-binary data elements (e.g. numbers, dates, text strings etc.) is that defined by ISO/IEC 8211 (i.e. ASCII, IRV of ISO/IEC 646). Some text string subfields may be encoded using an alternate character set. For this purpose two text string domain types are defined. These are “Basic Text” (used to encode alphanumeric identifiers, etc.) and “General Text” to handle certain attribute values (e.g. place names including accents and special characters).

Three lexical levels are defined for the encoding of these text string domains.

Level 0	ASCII text, IRV of ISO/IEC 646
Level 1	ISO 8859 part 1, Latin alphabet 1 repertoire (i.e. Western European Latin alphabet based languages.
Level 2	Universal Character Set repertoire UCS-2 implementation level 1 (no combining characters), Base Multilingual plane of ISO/IEC 10646 (i.e. including Latin alphabet, Greek, Cyrillic, Arabic, Chinese, Japanese etc.)

table 2.3

Basic text must always be encoded at lexical level 0. General text may be encoded at lexical level 0,1 or 2 depending on the type of field which holds the text string. Only the character-type subfields in the “Feature Record Attribute” [ATTF] and “Feature Record National Attribute” [NATF] fields may be encoded as general text. All other character-type subfields must be encoded as basic text.

General text in the ATTF field may be of level 0 or 1. All three levels can be used for the NATF field. In both cases a product specification may restrict the use of certain lexical levels. The lexical level used for these fields must be explicitly coded in the “ATTF Lexical Level” [AALL] and “NATF Lexical Level” [NALL] subfields of the “Data Set Structure Information” [DSSI] field. The default value for both subfields is zero. A thorough explanation of how the different levels are implemented is given in Annex B of part 3.

2.5 Field and Subfield termination

A Variable length subfield must be terminated by the “Unit Terminator” (UT). A variable length subfield is specified in the data structure by a format indicator without an extent (see clause 7.2.2.1). All S-57 fields (ISO/IEC 8211 data fields) must be terminated by the “Field Terminator” (FT).

When an alternate character set is used for a S-57 field, the UT and FT must be encoded at the lexical level specified for that field. Table 2.4 defines the terminators for each level.

Lexical level	UT	FT
level 0	(1/15)	(1/14)
level 1	(1/15)	(1/14)
level 2	(0/0) (1/15)	(0/0) (1/14)

table 2.4

2.6 Floating point values

Inspite of standards for the handling of binary encoded floating point values, different computer platforms often interpret floating point values differently. To avoid such problems, all floating point values in the binary implementation must be encoded as integers. In order to convert between the floating point and integer value, a multiplication factor is used. For coordinate and 3-D (sounding) values the multiplication factor is defined globally (see clause 3.2 and 3.3). Specific multiplication factors are defined on a per field basis for all other floating point values.

Encoding of floating point values is defined by the following algorithm:

*integer value = floating point value * multiplication factor*

The use of a multiplication factor for floating point values in the ASCII implementation is not mandatory; all floating point values can be encoded as R-types (see clause 7.2.2.1). If the multiplication factor is not used, its value must be set to 1.

2.7 Media and size restrictions

S-57 uses ISO/IEC 8211 as its encapsulation. ISO/IEC 8211 is media independant. Therefore, S-57 data can be stored or exchanged on any media. However, some applications might restrict the use of certain media types. The restriction of media type must be defined by the relevant product specification.

The only size restriction is a maximum field length of 10^9-1 bytes which is defined by ISO/IEC 8211 *). S-57 imposes no further restriction on the size of the various constructs in the data structure. However, some applications might limit the size of some or all constructs. These limitations must be defined by the relevant product specification.

*) The maximum field length is defined by the “size of field length field” in the ISO/IEC 8211 leader (LR RP 20, see Annex A).

2.8 Data quality

Data quality comprises the following:

- source of data;
- accuracy of data;
- up-to-dateness of data.

Data quality is considered to be meta information. As such, it can be encoded at three different levels (see clause 1.3). Data quality information is considered to be application specific. Therefore, rules for encoding data quality must be defined by the relevant product specification.

3. Meta record coding conventions

Meta data can be hierarchically encoded at three different levels (see clause 1.3). At the highest level in the hierarchy meta data is encoded by means of data set descriptive (meta) records. The following meta records are defined by S-57:

- Data Set General Information record
- Data Set Geographic Reference record
- Data Set History record
- Data set Accuracy record
- Catalogue Directory record

The use of these records within a data set must be specified in a product specification. However, the following clauses give some general guidelines which must be followed by the encoder.

3.1 Topology

Part 2 of this standard (Theoretical Data Model) defines, for vector data, the various levels of topological relationship. These levels are implemented as vector data structures.

Only one vector data structure may be used within a data set. The data structure used must be explicitly encoded in the “Data Structure” [DSTR] subfield of the “Data Set Structure Information” [DSSI] field. This field is part of the “Data Set General Information” record (see clause 7.3.1).
The options are:

CS {1} Cartographic spaghetti (see part 2, clause 2.2.1.1)

CN {2} Chain-node (see part 2, clause 2.2.1.2)

PG {3} Planar graph (see part 2, clause 2.2.1.3)

FT {4} Full topology (see part 2, clause 2.2.1.4)

NO {255} Topology is not relevant

If the data structure defined in the DSTR subfield is “Cartographic spaghetti” (CS), only vector records of type “isolated node” and “edge” may be present in the data set. The vector record type “connected node” is only permissible where the data structure has been defined as “Chain-node” (CN), "Planar graph" (PG) or “Full topology” (FT). The vector record type “face” is only permissible where the data structure has been defined as “Full topology” (FT).

3.2 Coordinate system, units and projection

This standard is intended to support the exchange of digital hydrographic data. As explained in part 2 (Theoretical Data Model), hydrographic data has both descriptive and spatial components. To facilitate the transfer of the spatial component (geometry), a coordinate system must be defined.

The “Data Set Geographic Reference” record (see clause 7.3.2) is used to exchange details about the coordinate system employed. These details contain two basic components:

- Coordinate units
- Projection

Both components are specified in detail in the clauses 3.2.1 and 3.2.2.

3.2.1 Coordinate units

Coordinates can be encoded in three different ways. Only one type of units is allowed within a data set. The type of unit is encoded in the "Coordinate Unit" [COUN] subfield of the "Data set Parameter" [DSPM] field. The options are:

LL {1} Latitude and longitude
EN {2} Easting/Northing
UC {3} Units on chart/map

Latitude and longitude units are degrees of arc. South and west are negative. Easting/Northing units are meters and units on the chart/map are milimeters.

Latitude and longitude	Degrees of arc
Easting/Northing	Meters
Units on chart/map	Milimeters

table 3.1

In the binary implementation coordinates are encoded as integer values. In order to convert floating-point coordinate values to integers (and vice-versa) the coordinate multiplication factor is used. The factor is defined by the encoder and held in the "Coordinate Multiplication Factor" [COMF] subfield. The COMF subfield applies to all coordinate fields defined in clause 7.7.1. The conversion algorithm is defined in clause 2.6.

3.2.2 Projection and registration control

When transforming units, other than latitude and longitude, into geographical positions (referenced to the earth' surface), the following data must be available:

- the chart/map projection employed, including the necessary parameters;
- a sufficient number of registration points (points for which both the unit coordinates and the geographical position are known).

The data indicated above must be encoded in "Data Set Projection" [DSPR] and "Data Set Registration Control" [DSRC] fields.

Up to 4 parameters can be specified in the "Data Set Projection" field. Possible values for these parameters are defined in table 3.2.

Name	Value for PROJ	Parameter 1	Parameter 2	Parameter 3	Parameter 4
Albert equal area	ALA {1}	Central meridian	Std. parallel nearer to equator	Std. parallel farther from equator	Parallel of origin
Azimuthal equal area	AZA {2}	Longitude of tangency	Latitude of tangency	–	–
Azimuthal equal distance	AZD {3}	Longitude of tangency	Latitude of tangency	–	–
Gnonomic	GNO {4}	Longitude of tangency	Latitude of tangency	–	–
Hotine oblique Mercator (rectified skew orthomorphic)	HOM {5}	Longitude of projection origin	Latitude of projection origin	Azimuth of skew X-axis at projection origin	Scale factor at projection origin
Lambert conformal conic	LCC {6}	Central meridian	Std. parallel nearer to equator	Std. parallel farther from equator	Parallel of origin
Lambert equal area	LEA {7}	Central meridian	–	–	–
Mercator	MER {8}	Central meridian	Latitude of true scale	Parallel of origin	–
Oblique Mercator	OME {9}	Longitude of reference point on great circle	Latitude reference point of great circle	Azimuth of great circle at ref. point	–
Orthographic	ORT {10}	Longitude of tangency	Latitude of tangency	–	–
Polar stereo graphic	PST {11}	Central meridian	Latitude of true scale	–	–
Polyconic	POL {12}	Central meridian	–	–	–
Transverse Mercator	TME {13}	Central meridian	Central scale factor	Parallel of origin	–
Oblique stereographic	OST {14}	Longitude of origin	Latitude of origin	Scale factor at origin	–

table 3.2

All latitudes and longitudes must be encoded as degrees of arc (South and West are negative).
If applicable, a false Easting and/or Northing can be specified in the FEAS and FNOR subfields of the “Data Set Projection” field. Both false Easting and Northing must be encoded as meters.

A total of 9 registration control points can be encoded in the “Data Set Registration Control” [DSRC] field. For each registration control point, both the unit values and the geographical position must be specified. Geographical positions of the control points must be encoded in latitude and longitude or Easting/Northing. The “Coordinate units for registration point” [CURP] subfield is used to indicate which units are used.

3.3 3-D (sounding) multiplication factor

In the binary implementation, 3-D sounding values are encoded as integers. In order to convert floating-point 3-D (sounding) values to integers (and vice-versa) a multiplication factor is used. The factor is defined by the encoder and held in the “3-D (sounding) Multiplication Factor” [SOMF] subfield. The SOMF subfield applies to the “3-D (sounding) Value” [VE3D] subfield of the “3-D Coordinate” [SG3D] field. The conversion algorithm is defined in clause 2.6.

3.4 Checksums

A “Cyclic Redundancy Check” (CRC) algorithm can be used to ensure that the data has not been corrupted during the exchange process. Different CRC algorithms can be used for different applications. The algorithm used is, therefore, described in the relevant product specification (see Appendix B – Product Specifications).
A CRC value for every file in an exchange set can be encoded in the “Catalogue Directory” [CATD] field, CRCS subfield.

4. Feature record coding conventions

4.1 General

An instance of a feature object class is implemented in the data structure as a feature record. Feature object classes are listed in Appendix A, IHO Object Catalogue. For each object class the IHO Object Catalogue defines permissible attributes.
The IHO Object Catalogue identifies 4 categories of object class:

- meta;
- cartographic;
- geo;
- collection.

Each category is implemented in the structure as a feature record and encoded in the same manner.

Feature records consist of the following fields:

- record identifier field;
- object identifier field;
- attribute fields;
- pointer control fields;
- pointer fields.

The “Pointer Control” fields are only used for updating. These fields are explained in chapter 8 (Updating). The other fields are discussed in the clauses 4.2 to 4.7.

In general, each instance of a feature object class requires one feature record. However, soundings are regarded as a special case in hydrography. In the interests of efficiency, soundings may be grouped into one feature record, provided that all the feature attributes and attribute values, except depth, are common to that group (see also clause 5.1.4.1).

4.2 Feature record identifier field

The identifier field consists of the following (groups of) subfields:

- record identifier [RCNM, RCID];
- object geometric primitive [PRIM];
- group [GRUP];
- object label/code [OBJL];
- record version [RVER];
- record update instruction [RUIN].

The record identifier is the basic identifier for the feature record and is defined in clause 2.2. The “Record Version” [RVER] and “Record Update Instruction” [RUIN] are used for updating. The mechanism for updating is explained in chapter 8. The other subfields are explained below.

4.2.1 Object geometric primitive [PRIM] subfield

The “Object Geometric Primitive” [PRIM] subfield is used to specify the geometric primitive of the encoded object. Permissible values are:

P {1} Point
L {2} Line
A {3} Area
N {255} Object does not directly reference any geometry

Allowable geometric primitives for object classes must be defined by the relevant product specification.

The PRIM subfield must be used to ensure the correct interpretation of the spatial record(s) to which the feature record refers. The “N” value is used for feature records which do not reference any spatial records (e.g. collection feature records).

The geometric primitive for all soundings (including grouped soundings) must be “P” (point).

4.2.2 Group [GRUP] subfield

The “Group” [GRUP] subfield is used to separate feature objects into groups. The definition of groups is dependent on the product specification (see Appendix B – Product Specifications). If a feature object does not belong to a group, the subfield must be left empty (see clause 2.1).

4.2.3 Object label/code [OBJL] subfield

The numeric object label/code of the object class from the IHO Object Catalogue is encoded in the “Object Label/Code” [OBJL] subfield.

4.3 Feature object identifier field

The feature object identifier field consists of the following subfields:

- producing agency [AGEN];
- feature Identification number [FIDN];
- feature Identification subdivision [FIDS].

The AGEN, FIDN and FIDS subfields are used as the identification (key) of a feature object (feature object identifier). The feature object identifier is also known as the “Long Name” [LNAM]. The LNAM subfield is used as a foreign pointer in the encoding of relationships between feature records (see chapter 6). Other applications for the feature object identifier, such as unique world-wide identification of feature objects, may be specified in the relevant product specification.

In the ASCII implementation, LNAM is a string of 17 characters (i.e. the concatenation of the AGEN, FIDN and FIDS subfields). The FIDN and FIDS subfields must be left filled with zeros (see clause 7.2.2.2).
In the binary implementation, LNAM is a bit string with a length of 64 bits (i.e. the “binary” concatenation of the AGEN, FIDN and FIDS subfields). Retrieval from the bit string of the individual subfield values must be resolved by the decoder.

4.3.1 Producing agency [AGEN] subfield

The allowable values for the "Producing Agency" [AGEN] subfield are defined in the IHO Object Catalogue. The IHO Object Catalogue contains a 2-character acronym and a corresponding integer value for each agency. If the producing agency is not listed, the AGEN subfield must be encoded as a missing subfield value (see clause 2.1).

4.3.2 Feature Object identification number and subdivision [FIDN, FIDS] subfield

The "Feature Object Identification Number" ranges from 1 to 2^{32}-2. The "Feature Object Identification Subdivision" ranges from 1 to 2^{16}-2. Both subfields are used to create an unique key for a feature object produced by the agency encoded in the AGEN subfield. The usage of the FIDN and FIDS subfields is not constrained and must be defined by the encoder.

4.4 Feature record attribute field

Attributes of feature objects must be encoded in the "Feature Record Attribute" [ATTF] field (see clause 7.6.3). The numeric attribute label/code of the attribute from the IHO Object Catalogue is encoded in the "Attribute Label/Code" [ATTL] subfield. In both the ASCII and binary implementations, the "Attribute Value" subfield [ATVL] must be a string of characters terminated by the subfield terminator (1/15). Lexical level 0 or 1 may be used for the general text in the ATTF field (see clause 2.4).

The IHO Object Catalogue (Appendix A) defines the valid attributes. For each attribute the IHO Object Catalogue defines the permissible attribute values.

An attribute is not allowed to repeat within one feature record.

4.5 Feature record national attribute field

National attributes of feature objects must be encoded in the "Feature Record National Attribute" [NATF] field (see clause 7.6.4). The numeric attribute label/code of the national attribute from the IHO Object Catalogue is encoded in the "Attribute Label/Code" [ATTL] subfield. In both the ASCII and binary implementations, the "Attribute Value" subfield [ATVL] must be a string of characters terminated by the appropriate subfield terminator (see clause 2.5). All lexical levels may be used for the general text in the NATF field (see clause 2.4).

The IHO Object Catalogue (Appendix A) defines the valid national attributes. For each national attribute the IHO Object Catalogue defines the permissible attribute values.

A national attribute is not allowed to repeat within one feature record.

4.6 Feature record to feature object pointer field

The "Feature Record to Feature Object Pointer" [FFPT] field is used to establish a relationship between feature objects. Relationships between feature objects are discussed in detail in chapter 6.

The main element of the pointer field is the LNAM subfield (see clause 4.3). The LNAM subfield contains the key of the feature object being referenced (foreign key). The "Relationship Indicator" [RIND] subfield can be used to qualify a relationship (e.g. master or slave relationship) or to add a stacking order to a relationship.

4.7 Feature record to spatial record pointer field

The “Feature Record to Spatial Record Pointer” [FSPT] field is used to link a feature record to its geometry.

The main element of the pointer field is the NAME subfield (see clause 2.2). The NAME subfield contains the key of the spatial record being referenced. The “Orientation” [ORNT] subfield, the “Usage Indicator” [USAG] subfield and the “Masking Indicator” [MASK] subfield are necessary for a correct interpretation of the spatial records being referenced.

The geometric primitive of a feature record determines the use of the pointer field. Geo, cartographic and meta feature records can be of geometric primitive point, line or area. The use of the pointers for these geometric primitives is specified in clauses 4.7.1 to 4.7.3. Feature records of geometric type point, line and area can only reference spatial records of type vector.

4.7.1 Feature record to spatial record pointer field — use by point features

In the chain-node, planar graph and full topology data structures, a point feature may reference isolated nodes or connected nodes. In the cartographic spaghetti data structure a point feature may only reference isolated nodes.
In general, feature records of type point can only reference one vector record. Multiple pointers to vector records are not allowed. An exception to this rule is a sounding feature record. Soundings are regarded as point features but are encoded in a special way (see clause 5.1.4.1).

The ORNT, USAG and MASK subfields must be set to “N” {255}.

4.7.2 Feature record to spatial record pointer field — use by line features

In order to facilitate the decoding of linear features comprising multiple edges, the vector records making up the linear feature must be referenced sequentially.

The direction in which an edge is to be interpreted for a particular linear feature may be important (e.g. for the symbolization of non-symmetric line styles). In such cases the direction of interpretation is indicated in the ORNT subfield (see figure 4.1). Permissible values are:

F {1} Forward.
R {2} Reverse.
N {255} Direction is not relevant.

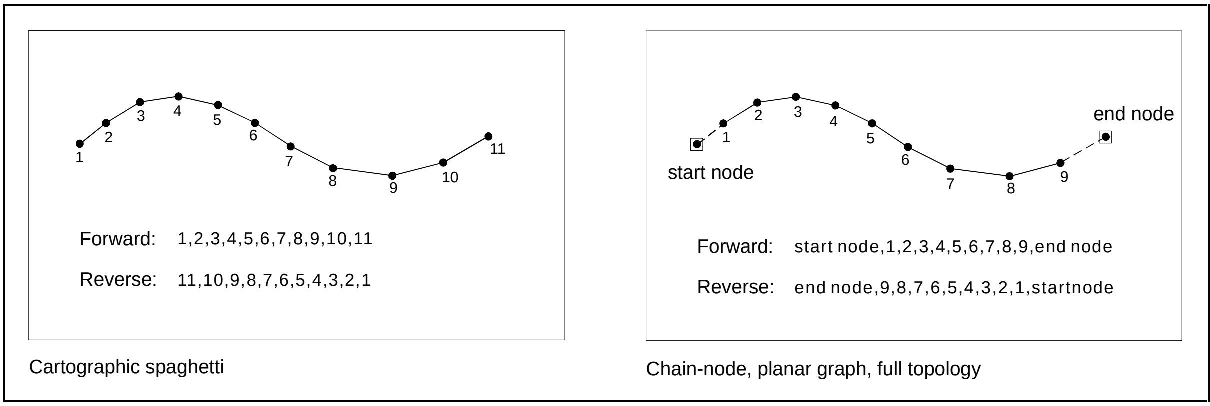

figure 4.1

The USAG subfield is set to “N” {255}. The MASK subfield specifies whether the referenced edge must be masked or not (see figure 4.2). Permissible values for the MASK subfield are:

M {1} Mask
S {2} Show
N {255} Masking is not relevant

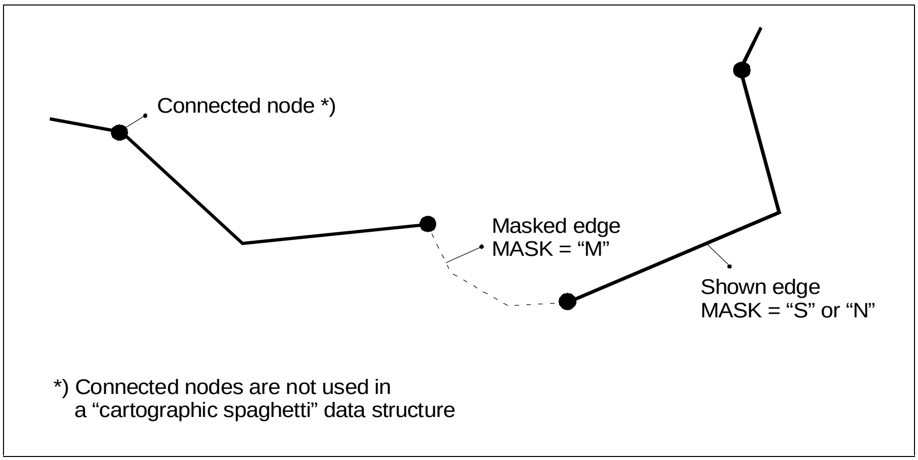

figure 4.2

4.7.3 Feature record to spatial record pointer field — use by area features

4.7.3.1 General

In the full topology data structure areas are encoded as one or more faces. The feature record references the faces and they in turn reference their bounding edges. For all other data structures faces are not valid. Therefore, areas are encoded as closed set of edges in the spaghetti data structure and closed sets of edges and connected nodes in the chain-node and planar graph data structures.

Closure of area and face boundaries must be explicit; implied closure is not allowed. This means that for the spaghetti data structure the first point of the first bounding edge must be identical to the last point of the last bounding edge. For the chain-node and planar graph data structures, the first and last edges bounding an area must meet at a common connected node. For full topology the first and last edges bounding a face must meet at a common connected node.

In order to facilitate the decoding of area data, vector records making up an area boundary must be referenced sequentially. The exterior boundary must be completely encoded before any interior boundaries and each interior boundary must be completed before encoding further interior boundaries.

4.7.3.2 Direction of area boundaries

Area outer boundaries must be encoded in a clockwise direction (i.e. so that the area lies to the right of the line). Area inner boundaries must be encoded in a counter clockwise direction (area to the right of the line). Consequently, the encoder must indicate in which direction (Forward or Reverse) the coordinates must be used to produce the clockwise (outer boundary) or counter clockwise (inner boundary) direction for that particular area (see figure 4.1 and 4.3).

The direction in which the edge is to be interpreted for a particular area is indicated in the “Orientation” [ORNT] subfield. Permissible values are:

F {1} Forward
R {2} Reverse

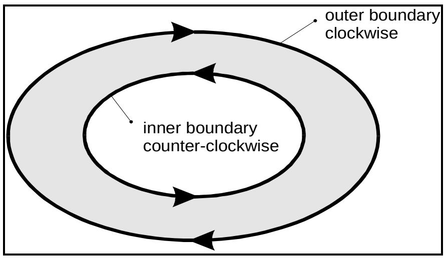

figure 4.3

4.7.3.3 Interior and exterior boundaries

In the case of areas consisting of one outer boundary and one or more non-intersecting inner boundaries (areas with holes), the “Usage Indicator” [USAG] subfield is used to distinguish between interior and exterior boundaries (see figure 4.4). This subfield is also used to indicate that an exterior boundary is part of the data limit. Permissible values are:

E {1} Exterior boundaries
I {2} Interior boundaries
C {3} Exterior boundary truncated by the data limit

A USAG subfield with value “C” must only be used when the feature is truncated by the data limit. For example, at a cell border in an ENC (see Appendix B.1 – ENC Application Profile). When the area feature limit coincides with the data limit the USAG subfield must be encoded as exterior (E).

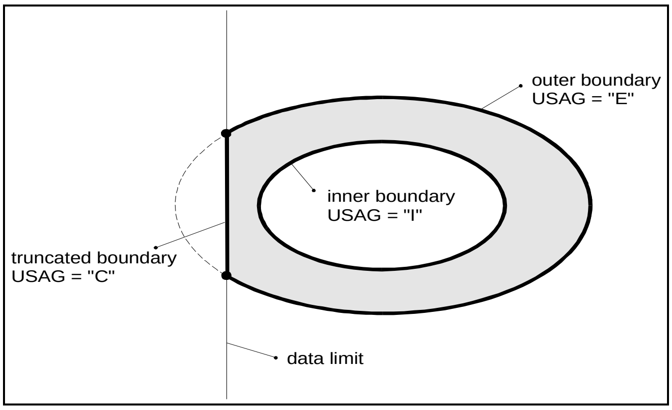

figure 4.4

An inner boundary which is truncated by the data limit becomes an outer boundary after the truncation (see figure 4.5).

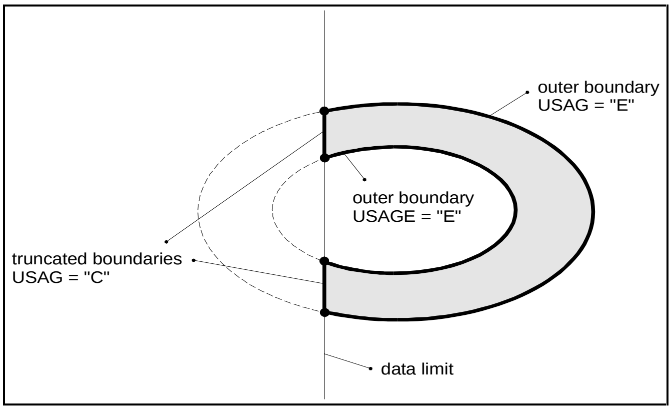

figure 4.5

4.7.3.4 Masking of area boundaries

Under certain circumstances it may be necessary to suppress the symbolization of one or more edges which define the inner or outer boundary of an area. Suppression of the symbolization can be controlled by using the “Masking Indicator” [MASK] subfield (see figure 4.2). Permissible values are:

M {1} Mask
S {2} Show
N {255} Masking is not relevant

5. Spatial record coding conventions

This Standard defines three types of spatial record. These are vector, raster and matrix. The spatial record types are further defined in the clauses 5.1 to 5.3.

5.1 Vector records

A vector record can be of type isolated node, connected node, edge or face. The geometry of soundings is considered to be a special case of an isolated node (see clause 5.1.4.1).
Vector records consist of the following fields:

- record identifier field;
- attribute field;
- pointer control field;
- pointer field;
- coordinate control field;
- coordinate fields *)

*) This standard defines different types of coordinate fields which are mutually exclusive within one vector record (see clause 5.1.4).

The “Pointer Control” field and the “Coordinate Control” field are only used for updating. The mechanism for updating is explained in chapter 8. The other fields are discussed below.

A detailed structure description of the vector record is given in clause 7.7.1.

5.1.1 Vector record identifier field

The vector record identifier field holds the record identifier (key) for that vector record (see clause 2.2). It is also used to differentiate between the various types of vector records. For this purpose the “Record Name” [RCNM] subfield is used. Depending on the type of vector record, the RCNM subfield can take the following values:

VI {110} Isolated node
VC {120} Connected node
VE {130} Edge
VF {140} Face

5.1.2 Vector record attribute field

Attributes of vector records must be encoded in the “Vector Record Attribute” [ATTV] field (see clause 7.7.1.2). The numeric attribute label/code of the attribute from the IHO Object Catalogue is encoded in the “Attribute Label/Code” [ATTL] subfield. In both the ASCII and binary implementations, the “Attribute Value” subfield [ATVL] must be a string of characters terminated by the subfield terminator (1/15).

The IHO Object Catalogue (Appendix A) defines the valid attributes for vector objects. For each attribute the IHO Object Catalogue defines the permissible attribute values.

An attribute is not allowed to repeat within one vector record.

5.1.3 Vector record pointer field

The “Vector Record Pointer” [VRPT] field is used to maintain the correct topological relationships within the data.
This field must be used in the chain-node, planar graph and full topology data structures for the edge vector records. This field must also be used in the full topology data structure for the isolated node and face vector records. It must not be used for the connected node vector record.
In the spaghetti data structure this field is meaningless and, therefore, must not be used.

The use of this field for the isolated node, edge and face vector records is explained in the clauses 5.1.3.1 to 5.1.3.3.

5.1.3.1 Vector record pointer field — use by isolated nodes

In the full topology data structure an isolated node may reference the face in which it is contained (i.e. pointer from isolated node to face). For this purpose the “Topology Indicator” [TOPI] subfield is used. The following value must be used:

F {5} Containing face

The ORNT, USAG and MASK subfields must be set to “N” {255}.

5.1.3.2 Vector record pointer field — use by edges

In the chain-node, planar graph and full topology data structures the beginning and end of an edge are explicitly coded as connected nodes. The connected nodes are referenced by the edge (i.e. pointer from edge to connected node). Edges which close on themselves (loop feature) must reference the same connected node twice.
In the full topology data structure edges must also reference the faces to their left and right.

All references are mandatory. Omission of these references results in a corruption of the topological structure.

The pointer type is identified by the value of the “Topology Indicator” [TOPI] subfield. The following values are permitted for an edge vector record:

B {1} Beginning node
E {2} End node
S {3} Left face
D {4} Right face

All values are relative to the coding direction (see figure 5.1). References must be made in the sequence indicated above (beginning node, end node, left face, right face).

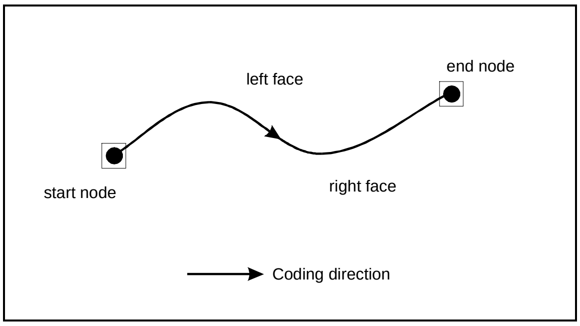

figure 5.1

The ORNT, USAG and MASK subfields must be set to “N” {255}.

5.1.3.3 Vector record pointer field — use by faces

A face vector record can only reference edges. A face vector record references an edge in the same way as an area feature record references an edge. The mechanism is described in clause 4.7.3. The direction in which an edge must be interpreted is encoded in the “Orientation” [ORNT] subfield. The “Usage Indicator” [USAG] subfield specifies whether an edge is part of the interior or exterior boundary of the face or coincides with the data limit. The “Masking Indicator” [MASK] subfield specifies whether the referenced edge must be masked or not.

For a face vector record, the following “Orientation” [ORNT], “Usage Indicator” [USAG] and “Masking Indicator” [MASK] subfield values are permitted:

ORNT	**F**	{1}	Forward
	R	{2}	Reverse
USAG	**E**	{1}	Exterior boundaries
	I	{2}	Interior boundaries
	C	{3}	Exterior boundary truncated by the data limit
MASK	**M**	{1}	Mask
	S	{2}	Show
	N	{255}	Masking is not relevant

The TOPI subfield must be set to “N” {255}

References from face to edge are mandatory for the full topology data structures. Omission of these references results in a corruption of the topological structure.

5.1.4 Coordinate fields

The actual spatial component (the geometry) of the exchanged vector data is encoded in the coordinate fields. This standard defines the following coordinate fields for vector data:

- 2-D coordinate field;
- 3-D coordinate field (sounding array);
- Arc/Curve coordinate fields.

The coordinate fields above are mutually exclusive within one vector record. The use of these fields depends on the type of vector record in which they are encoded. An explanation of the sounding, isolated node, connected node and edge vector records is given in clauses 5.1.4.1 to 5.1.4.4.

Soundings are not explicitly defined by this standard as a separate vector record type. They are considered to be a special case of isolated nodes. However, they are encoded differently and are, therefore, discussed separately.

A face vector record does not contain coordinate fields. It only references edge vector records (see clause 5.1.3.3).

This Standard allows for coordinates to be stored using a number of different units. Refer to clause 3.2 for details on how to encode coordinates.

5.1.4.1 Coordinate fields — use by soundings

In the interests of efficiency, soundings may be grouped into one vector record, provided that all spatial attributes and attribute values are common to that group (see also clause 4.1). A special construct for the encoding of soundings is provided, called a 3-D Coordinate or sounding array field, with the tag SG3D. Within this field the sounding value is held as the third component of repeating Y-coordinate, X-coordinate and depth triplets.

5.1.4.2 Coordinate fields — use by isolated nodes

Coordinates of isolated nodes must be encoded in the “2-D Coordinate” [SG2D] field and will comprise a single coordinate pair (see clause 7.7.1.6).

5.1.4.3 Coordinate fields — use by connected nodes

Coordinates of connected nodes must be encoded in the “2-D Coordinate” [SG2D] field and will comprise a single coordinate pair (see clause 7.7.1.6).

5.1.4.4 Coordinate fields — use by edges

Coordinates of edges may be encoded using either the “2-D Coordinate” [SG2D] field or the “Arc Curve Definition” [ARCC] field. The SG2D field holds repeating pairs of coordinates implicitly joined by straight lines. The ARCC field provides the means for defining mathematically derivable arcs and curves of four types. The SG2D and ARCC fields are mutually exclusive.

In chain-node, planar graph and full topology data structures the beginning and end of an edge are explicitly coded as connected nodes. The geometry of the connected node is not part of the edge. Edges directly reference their beginning and end nodes via the vector record pointer (see clause 5.1.3.2). The first and last points in the edge coordinate field are connected to the connected nodes by implied straight lines. A straight line between two connected nodes must be encoded as an edge with a reference to both connected nodes (VRPT) but no coordinate geometry (i.e. no SG2D or ARCC coordinate fields).
The type of interpolation used for an arc/curve representation is indicated in the “Arc/Curve Type” [ATYP] subfield (see clause 7.7.1.8). The options are:

Arc Representations: (see clause 7.7.1.9 and 7.7.1.10)

C {1} Arc 3 Point Centre: described with 3 points; a starting point on the arc [STPT], centre point [CTPT], end point [ENPT] forming the end vector. The starting point forms both the start vector and the defining point for the radius. The arc must be drawn in *clockwise direction* about the centre point until the end vector is encountered. The arc ENPT must be located on the arc (see figure 5.2).

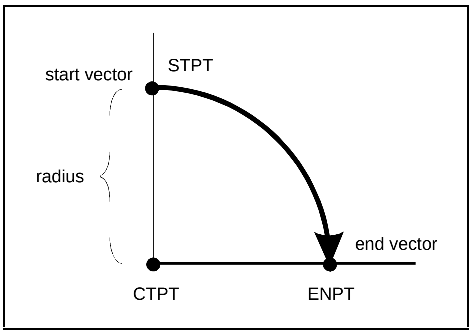

figure 5.2

E {2} Elliptical Arc: described with 5 points; starting point [STPT] forming the start vector, centre point [CTPT], end point [ENPT] forming the end vector, conjugate diameter point (CDP) on the major axis [CDPM] of the ellipse, and CDP on the minor axis [CDPR] of the ellipse. The angle between the major and minor axis of the ellipse must be assumed to be 90°. The ellipse must be drawn in a *clockwise direction* about the centre point passing through the CDP constructed on each side of the centre point on the axes. The arc STPT and ENP must be located on the arc (see figure 5.3).

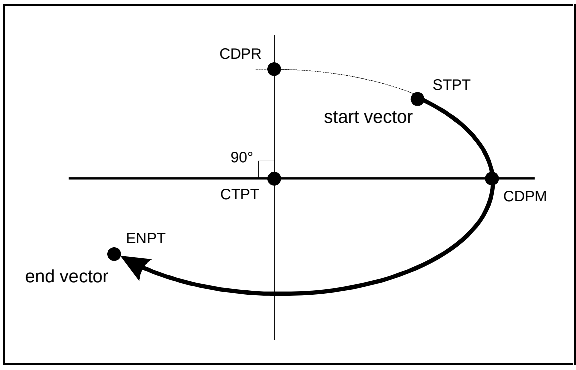

figure 5.3

Curve Representations: (see clause 7.7.1.11)

The encoder can describe a very complex curve using straight line segments and control points of sufficiently high resolution to appear to the user as a very smooth curve. The advantage of this approach is the simplicity of the data structure; the disadvantage is the data volume (i.e. number of coordinate pairs) required to provide the smooth appearance. In addition, the smoothness may be lost if the data were to be used at scales substantially larger than the encoder intended.

Conversely, the encoder can describe a very complex curve using a specific polynomial description which, if accurately regenerated, will appear to the user as a very smooth curve. The advantage of this approach is the lower data volume relative to the method desribed above; the disadvantage is that the user may not be capable of regenerating the curve as intended.

Clause 7.7.1.8 does not describe the underlying polynomials as the implementation may vary significantly amongst various vendor systems. The information required to adequately describe the polynomial characteristics is provided in a structured form. Thus, when specifying to the user that a polynomial based regeneration is desired, the ARCC field must be used in conjunction with the CT2D field for control points. Within the ARCC field, the ATYP (Arc/Curve Type), SURF (Construction Surface), ORDR (Curve Order), and RESO (Point Resolution) subfields must be used to accurately describe the characteristics *that the encoder* intends to be applied to the control points of CT2D.

If the user applies these characteristics in any way other than that intended by the encoder, the values of the associated attributes (e.g. quality attributes) *may* not apply.

A more thorough understanding of the underlying mathematics may be found in any of a number of texts on computer graphics or by consulting any of the several computer graphics drawing standards. A more thorough understanding of a particular vendor implementation may be sought from the specific vendor.

The three implemented curve types have the following implementation details:

U {3} Uniform B-Spline

a) use one ARCC field to describe characteristics with one or more CT2D fields to describe all control points.

B {4} Piecewise Bezier

a) use one ARCC field to describe characteristics with one or more CT2D fields to describe all control points.

N {5} Non-Uniform Rational B-Spline

a) use one ARCC field to describe the characteristics of *each set* of polynomials with one or more CT2D fields to describe the control points *for the specified set*, and

b) use multiple pointers to edge vector records in the FSPT field to group the sets into a complete representation.

5.2 Raster record coding conventions

To be defined.

5.3 Matrix record coding conventions

To be defined.

6. Relationship coding

Relationships between records can be encoded in three ways:

- by using a “Catalogue Cross Reference” record;
- by using a collection feature record;
- by defining a nominated “master” feature record.

These methods are described in clause 6.1 to 6.3 respectively. Additional rules for relationship coding may be defined by the relevant product specification.

6.1 Catalogue Cross Reference record

The Catalogue Cross Reference record can be used to link records of any type within an exchange set. The two records are identified by foreign pointers (see clause 2.2) held in the NAM1 and NAM2 subfields. The exact nature of the relationship can only be indicated by use of the “Comment” [COMT] subfield.

Only one relationship can be encoded in one Catalogue Cross Reference field.

6.2 Collection feature record

A collection feature record is the data structure implementation of a collection object. A collection feature record is formed in the same way as other feature records (see chapter 4).

The collection object classes are defined in the IHO Object Catalogue (Appendix A).

A collection feature record may only reference feature objects. It must not reference any spatial records. The “Object geometric primitive” [PRIM] subfield must, therefore, be “N” {255}.

The relationship is encoded using the feature record to feature object pointer field. This field holds the LNAM foreign pointer (see clause 4.3) of one feature object. A collection feature record must reference at least two other feature objects and must not reference itself. Collection feature records may reference other collection feature objects.

The “Relationship indicator” [RIND] subfield is used to indicate the nature of the relationship. It may have one of the following values:

M {1} master
S {2} slave
P {3} peer

Additional values may be defined by the relevant product specification.

There can be only one master (M) relationship per collection feature record. All remaining relationships from that collection feature record must be slave (S). If one relationship is defined as peer (P) all other relationships from that collection feature record must also be defined as peer (P).

All feature objects referenced by a collection feature record are related in the same way (i.e. that defined by the collection object class).

A collection object may have attributes. The allowable attributes for each collection object class are defined in the Object Catalogue. The use and meaning of these attributes must be defined by the appropriate product specification.

6.3 Nominated “master” feature record

In order to facilitate efficient coding, hierarchical relationships (i.e. master to slave) may be encoded by nominating one feature record as the “master” of the relationship (for example, a buoy might be considered the master and the topmark, light and fog signal might be considered its slaves). This master feature record must carry a feature record to feature object pointer field for each related slave object. This field holds the LNAM foreign pointer (see clause 4.3) of one feature object.

In all other respects the master feature record is the same as other feature records, it may have attributes and must reference at least one spatial record (see chapter 4).

This relationship is always master to slave; the RIND subfield must contain the value “S” {2}. This mechanism cannot be used for encoding peer to peer relationships.

Master feature records may reference other master feature objects but must not reference themselves.

7. Structure implementation

7.1 Introduction

This chapter specifies the structure of an exchange set at the record and field levels. It further specifies the contents of the physical constructs required for their implementation as ISO/IEC 8211 data records, fields, and subfields. The grouping of records into ISO/IEC 8211 files is considered application specific and is, therefore, described in the relevant product specification (see Appendix B – Product Specifications).

7.2 Notations used in this clause

The specification of the structure is given as a tree structure diagram which comprises the names, linkages and repetition factors of the physical constructs. The detailed specifications of fields and subfields are given in tabular form.

7.2.1 Tree structure diagrams

The structure of a record is an ordered rooted tree, represented as follows:

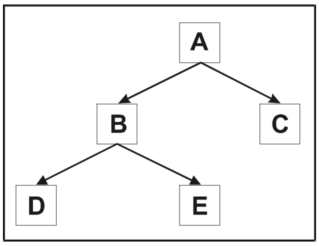

figure 7.1

Where A is the root node and parent of node B and node C. Node B is the root of a sub-tree and the parent of nodes D and E. Nodes are also referred to as the offspring of their parents. E.g. node B is the offspring of node A.

The tree structure diagrams must be interpreted in a preordered traversal sequence (top down, left branch first).

For ease of annotation these diagrams are presented vertically in this standard using ASCII characters. In this notation the above diagram becomes:

```
A
|
|--<r>----  B
|           |
|           *---D
|           |
|           *---E
|
|--<R>---   C
```

figure 7.2

Where: <r> is a sub-tree repetition factor (if missing, r=1)
<R> implies indefinite repetition

* implies either sub-tree D or E but not both.

In this way the preorder traversal sequence becomes “top down”.

The tree structure diagram specifies for each field: the ISO/IEC 8211 field tag, an indication of the field structure and a field name, i.e.

[field tag] [structure] [field name]

where *[structure]* has the values:

<n>	implies an n-tuple (1-D array) containing non-repeating subfields
<m*n>	implies an m by n array with m rows and n colums
<*n>	implies a n-column table with indefinitely repeating rows

The tree structure diagrams define which fields are allowed to repeat. However, within a record, the degree of repetition of fields will depend on the data that is being encoded. In some cases a particular field may not be required and so will be absent (see clause 2.1). However, in all cases, the pre-order traversal sequence of a data record will be the same as shown in the generic tree structure diagram for that record type.

N.B. The notation of the tree structure diagrams is described in detail in ISO/IEC 8211:1994 Annex B.1

7.2.2 Field tables

Each table is preceeded by a row in bold outline indicating the field name and field tag. The body of the table specifies the subfield names and labels as well as both the ASCII and binary (Bin) alternate ISO 8211 formats and the S-57 ASCII domain (Dom). The subfield specification may include a required value or range constraint. The following is an example of a field table based on the Data Set Identification field (not all subfields are displayed, refer to 7.3.1.1 for the complete description)

Field Tag: DSID [Upd] *)	Field Name: Data Set Identification

Subfield name	Label	Format ASCII	Bin	Dom	Subfield content and specification
Record Name	RCNM	A(2)	b11	an	"DS" {10} **)
Record Identification Number	RCID	I(10)	b14	dg	Range: 1 to 2^{32}-2
Exchange Purpose	EXPP	A(1)	b11	an	"N" {1} - Data set is New "R" {2} - Data set is a revision to an existing one
Intended usage	INTU	I(1)	***)	bt	A numeric value indicating the intended usage for which the data has been compiled (see Appendix B - Product Specifications)
Data set name	DSNM	A( )	***)	bt	A String indicating the data set name (see Appendix B - Product Specifications)
Edition number	EDTN	A( )	***)	bt	A string indicating the edition number (see Appendix B - Product Specifications)
...					

table 7.1

*) [Upd] indicates that the field is only used for updating (for the DSID field this is used as an example)
**) Required ASCII values are enclosed in double quotes and the values to be encoded in binary are enclosed in {...}
***) When a binary format is not specified the ASCII format applies.

Where: - **Label** is the ISO/IEC 8211 subfield label, present only in the data descriptive record and required to identify the subfields within a field. A label preceded by “*” signifies that the subfield and the subsequent ones, repeat within the field. This, therefore, indicates the presence of a 2-D array or table for which the subfield labels provide the column headings (the vector labels of a cartesian label).

- **Format** is the ISO/IEC 8211 ASCII or binary subfield data format (see table 7.2)

- **Dom** is the S-57 ASCII domain (see table 7.3)

7.2.2.1 Data format

Subfield data formats are specified by ISO/IEC 8211. The allowable data formats are as follows:

Format	Precision = w	Data type
A	*)	Character data
I	*)	Implicit point representation
R	*)	Explicit point representation
B	**)	Bit string
@		subfield label is a row heading for a 2-D array or table of known length
b1w	1,2,4 ***)	unsigned integer
b2w	1,2,4 ***)	signed integer

table 7.2

*) An extent of X(n) indicates a fixed length subfield of length n (in bytes). An extent of X( ) indicates a variable length subfield terminated by the appropriate delimiter (see clause 2.5).

**) The width of a fixed length bit subfield must be specified in bits. If necessary, the last byte of a fixed length bit subfield must be filled on the right with binary zero's.

***) In the binary form, numerical data forms are constrained by the precision of the ISO/IEC 8211 binary format.

Where:	Precision	is the width of the data items in bytes
	w	is a permitted value of precision
	unsigned integer	is a binary integer
	signed integer	is a two's complement binary integer

Binary values and multi-byte character codes (see clause 2.4 and Annex B) must be stored in the "least significant byte first" (LSBF or "little-endian") order. LSBF is an ordering of bytes in which the least significant byte is placed closest to the beginning of a file.

7.2.2.2 Permitted S-57 (ASCII) Data domains

The domain for ASCII data is specified by a domain code. The following domain codes are used in the field tables:

Domain code	Domain description
bt	Basic text (see clause 2.4)
gt	General text (see clause 2.4)
dg	digits; 0-9, right-adjusted and zero filled left (e.g. A(2) "03")
date	a date subfield in the form: YYYYMMDD (e.g. "19960101")
int	integer; ISO 6093 NR1, SPACE, "+", "-", 0-9, right-adjusted and zero filled left (e.g. I(5) “00015”)
real	real number; ISO 6093 NR2, SPACE, "+", "-", ".", 0-9
an	alphanumerics; A-Z, a-z, 0-9, "*", "?"
hex	hexadecimals; A-F, 0-9

table 7.3

7.3 Data set descriptive records

7.3.1 Data set general information record structure

```
Data Set General Information record
  |
  |--0001 (1) ISO/IEC 8211 Record Identifier
      |
      |--DSID (16) - Data Set Identification field
          |
          |--DSSI (11) - Data Set Structure information field
```

7.3.1.1 Data set identification field structure

Field Tag: DSID	Field Name: Data Set Identification

Subfield name	Label	Format ASCII	Format Bin	Dom	Subfield content and specification
Record name	RCNM	A(2)	b11	an	"DS" {10}
Record identification number	RCID	I(10)	b14	int	Range: 1 to 2^{32}-2
Exchange purpose	EXPP	A(1)	b11	an	"N" {1} Data set is New "R" {2} Data set is a revision to an existing one
Intended usage	INTU	I(1)	b11	int	A numeric value indicating the intended usage for which the data has been compiled (see Appendix B - Product Specifications)
Data set name	DSNM	A( )		bt	A string indicating the data set name (see Appendix B - Product Specifications)
Edition number	EDTN	A( )		bt	A string indicating the “edition number” (see Appendix B - Product Specifications)

Update number	UPDN	A( )		bt	A string indicating the “update number” (see Appendix B - Product Specifications)
Update application date	UADT	A(8)		date	All updates dated on or before this date must have been applied (see Appendix B - Product Specifications)
Issue date	ISDT	A(8)		date	Date on which the data was made available by the data producer (see Appendix B - Product Specifications)
Edition number of S-57	STED	R(4)		real	"03.1" Edition number of S-57
Product Specification	PRSP	A(3)	b11	an	"ENC" {1} Electronic Navigational Chart "ODD" {2} IHO Object Catalogue Data Dictionary (see 1.4.1)
Product specification description	PSDN	A( )		bt	A string identifying a non standard product specification (see 1.4.1)
Product specification edition number	PRED	A( )		bt	A string identifying the edition number of the product specification (see 1.4.1)
Application profile identification	PROF	A(2)	b11	an	"EN" {1} ENC New "ER" {2} ENC Revision "DD" {3} IHO Data dictionary (see 1.4.2)
Producing agency	AGEN	A(2)	b12	an	Agency code (see IHO Object Catalogue)
Comment	COMT	A( )		bt	A string of characters

table 7.4

7.3.1.2 Data set structure information field structure

Field Tag: DSSI	Field Name: Data Set Structure information

Subfield name	Label	Format ASCII	Bin	Dom	Subfield content and specification
Data structure	DSTR	A(2)	b11	an	"CS" {1} Cartographic spaghetti "CN" {2} Chain-node "PG" {3} Planar graph "FT" {4} Full topology "NO" {255} Topology is not relevant (see 3.1 and part 2 Theoretical Data Model)
ATTF lexical level	AALL	I(1)	b11	int	Lexical level used for the ATTF fields (see 2.4)
NATF lexical level	NALL	I(1)	b11	int	Lexical level used for the NATF fields (see 2.4)
Number of meta records	NOMR	I( )	b14	int	Number of meta records in the data set
Number of cartographic records	NOCR	I( )	b14	int	Number of cartographic records in the data set
Number of geo records	NOGR	I( )	b14	int	Number of geo records in the data set
Number of collection records	NOLR	I( )	b14	int	Number of collection records in the data set
Number of isolated node records	NOIN	I( )	b14	int	Number of isolated node records in the data set
Number of connected node records	NOCN	I( )	b14	int	Number of connected node records in the data set
Number of edge records	NOED	I( )	b14	int	Number of edge records in the data set
Number of face records	NOFA	I( )	b14	int	Number of face records in the data set

table 7.5

7.3.2 Data set geographic reference record structure

```
Data Set Geographic Reference record
    |
    |--0001 (1) ISO/IEC 8211 Record Identifier
        |
        |--DSPM (12)    Data Set Parameter field
            |
            |--DSPR (9)      Data Set Projection field
            |
            |--DSRC (*8)Data Set Registration control field
```

7.3.2.1 Data set parameter field structure

Field Tag: DSPM	Field Name: Data Set Parameter

Subfield name	Label	Format ASCII	Format Bin	Dom	Subfield content and specification
Record name	RCNM	A(2)	b11	an	"DP" {20}
Record identification number	RCID	I(10)	b14	int	Range: 1 to 2^{32}-2
Horizontal geodetic datum	HDAT	I(3)	b11	int	Value taken from the attribute HORDAT (see Appendix A - Object Catalogue)
Vertical datum	VDAT	I(2)	b11	int	Value taken from the attribute VERDAT (see Appendix A - Object Catalogue)
Sounding datum	SDAT	I(2)	b11	int	Value taken from the attribute VERDAT (see Appendix A - Object Catalogue)
Compilation scale of data	CSCL	I( )	b14	int	The modulus of the compilation scale. For example, a scale of 1:25000 is encoded as 25000
Units of depth measurement	DUNI	I(2)	b11	int	Value taken from the attribute DUNITS (see Appendix A - Object Catalogue)
Units of height measurement	HUNI	I(2)	b11	int	Value taken from the attribute HUNITS (see Appendix A - Object Catalogue)
Units of positional accuracy	PUNI	I(2)	b11	int	Value taken from the attribute PUNITS (see Appendix A - Object Catalogue)
Coordinate units	COUN	A(2)	b11	an	Unit of measurement for coordinates "LL" {1} Latitude/Longitude "EN" {2} Easting/Northing "UC" {3} Units on the chart/map (see 3.2.1)
Coordinate multiplication factor	COMF	I( )	b14	int	Floating-point to integer multiplication factor for coordinate values (see 3.2.1)
3-D (sounding) multiplication factor	SOMF	I( )	b14	int	Floating point to integer multiplication factor for 3-D (sounding) values (see 3.3)
Comment	COMT	A( )		bt	A string of characters

table 7.6

7.3.2.2 Data set projection field structure

Field Tag: DSPR	Field Name: Data Set Projection

Subfield name	Label	Format ASCII	Format Bin	Dom	Subfield content and specification
Projection	PROJ	A(3)	b11	an	Projection code taken from table 3.2 (see 3.2.2)
Projection parameter 1	PRP1	R( )	b24 *)	real	Content of parameter 1 is defined by the value of PROJ (see 3.2.2)
Projection parameter 2	PRP2	R( )	b24 *)	real	Content of parameter 2 is defined by the value of PROJ (see 3.2.2)
Projection parameter 3	PRP3	R( )	b24 *)	real	Content of parameter 3 is defined by the value of PROJ (see 3.2.2)
Projection parameter 4	PRP4	R( )	b24 *)	real	Content of parameter 4 is defined by the value of PROJ (see 3.2.2)
False Easting	FEAS	R( )	b24 *)	real	False easting of projection in meters (see 3.2.2)
False Northing	FNOR	R( )	b24 *)	real	False northing of projection in meters (see 3.2.2)
Floating point multiplication factor	FPMF	I( )	b14	int	Floating point to integer multiplication factor for projection parameters (see 2.6)
Comment	COMT	A( )		bt	A string of characters

table 7.7

*) use FPMF subfield to convert integer (b24) to floating point (see clause 2.6).

7.3.2.3 Data set registration control field structure

Field Tag: DSRC	Field Name: Data Set Registration Control

Subfield name	Label	Format ASCII	Format Bin	Dom	Subfield content and specification
Registration point ID	*RPID	A(1)	b11	dg	Range: 1 to 9 (see 3.2.2)
Registration point Latitude or Northing	RYCO	R( )	b24 *)	real	Latitude or Northing of registration point. Latitude in degrees of arc, Northing in meters (see 3.2.2)
Registration point Longitude or Easting	RXCO	R( )	b24 *)	real	Longitude or Easting of registration point. Longitude in degrees of arc, Easting in meters (see 3.2.2)
Coordinate units for registration point	CURP	A(2)	b11	an	"LL" {1} Latitude and Longitude "EN" {2} Easting and Northing
Floating point multiplication factor	FPMF	I( )	b14	int	Floating point to integer multiplication factor for Registration points RYCO and RXCO (see 2.6)
Registration point X-value	RXVL	R( )	b24	real	Unit X-value for registration point. Floating-point to integer conversion is defined by the COMF subfield of the DSPM field (see 3.2.2)
Registration point Y-value	RYVL	R( )	b24	real	Unit Y-value for registration point. Floating-point to integer conversion is defined by the COMF subfield of the DSPM field (see 3.2.2)
Comment	COMT	A( )		bt	A string of characters

table 7.8

*) use FPMF subfield to convert integer (b24) to floating point (see clause 2.6)

7.3.3 Data set history record structure

```
Data Set History record
   |
   |--0001 (1) ISO/IEC 8211 Record Identifier
       |
       |--DSHT (8)  - Data Set History field
```

Field Tag: DSHT	Field Name: Data Set History

Subfield name	Label	Format ASCII	Format Bin	Dom	Subfield content and specification
Record name	RCNM	A(2)	b11	an	"DH" {30}
Record identification number	RCID	I(10)	b14	int	Range: 1 to 2^{32}-2
Producing agency code	PRCO	A(2)	b12	an	Agency code (see IHO Object Catalogue)
Earliest source date	ESDT	A(8)		date	Date of the oldest source material within the coverage area
Latest source date	LSDT	A(8)		date	Date of the newest source material within the coverage area
Data collection criteria	DCRT	A( )		bt	A string indicating the criteria used for data collection
Compilation date	CODT	A(8)		date	Compilation date
Comment	COMT	A( )		bt	A string of characters

table 7.9

7.3.4 Data set accuracy record structure

```
Data Set Accuracy record
   |
   |--0001 (1) ISO/IEC 8211 Record Identifier
       |
       |--DSAC (7)  - Data Set Accuracy field
```

Field Tag: DSAC	Field Name: Data Set Accuracy

Subfield name	Label	Format ASCII	Format Bin	Dom	Subfield content and specification
Record name	RCNM	A(2)	b11	an	"DA" {40}
Record identification number	RCID	I(10)	b14	int	Range: 1 to 2^{32}-2
Absolute positional accuracy	PACC	R( )	b14 *)	real	The best estimate of the positional accuracy of the data. The expected input is the radius of the two-dimensional error.
Absolute horizontal/vertical measurement accuracy	HACC	R( )	b14 *)	real	The best estimate of the horizontal/vertical measurement accuracy of the data. The error is assumed to be both positive and negative. Subfield must be used to indicate the accuracy of horizontal/vertical measurements. Accuracy of soundings is encoded in the SACC subfield.

Absolute sounding accuracy	SACC	R( )	b14 *)	real	The best estimate of the sounding accuracy of the data. The error is assumed to be both positive and negative. Subfield must be used to indicate the vertical accuracy of soundings. Accuracy of horizontal/ vertical measurements is encoded in the HACC subfield.
Floating point multiplication factor	FPMF	I( )	b14	int	Floating point to integer multiplication factor for accuracy values (see 2.6)
Comment	COMT	A( )		bt	A string of characters

table 7.10

*) use FPMF subfield to convert integer (b14) to floating point (see clause 2.6).

7.4 Catalogue records

7.4.1 Catalogue directory record structure

```
Catalogue Directory record
    |
    |--0001 (1) ISO/IEC 8211 Record Identifier
        |
        |--CATD (12)     - Catalogue Directory field
```

Field Tag: CATD	Field Name: Catalogue Directory

Subfield name	Label	Format ASCII	Bin	Dom	Subfield content and specification
Record name	RCNM	A(2)		an	"CD"
Record identification number	RCID	I(10)		int	Range: 1 to 2^{32}-2
File name	FILE	A( )		bt	A string indicating a valid file name (see Appendix B - Product Specifications)
File long name	LFIL	A( )		bt	A string indicating the long name of the file (see Appendix B - Product Specifications)
Volume	VOLM	A( )		bt	A string indicating a valid volume label for the transfer media on which the file, indicated by the FILE subfield, is located. (see Appendix B - Product Specifications)
Implementation	IMPL	A(3)		an	"ASC" File is a S-57 ASCII implementation "BIN" File is a S-57 binary implementation Codes for non ISO/IEC 8211 files within an exchange set may be defined by a Product Specification (see Appendix B)
Southernmost latitude	SLAT	R( )		real	Southernmost latitude of data coverage contained in the file indicated by the FILE subfield. Degrees of arc, south is negative
Westernmost longitude	WLON	R( )		real	Westernmost longitute of data coverage contained in the file indicated by the FILE subfield. Degrees of arc, west is negative
Northernmost latitude	NLAT	R( )		real	Northernmost latitude of data coverage contained in the file indicated by the FILE subfield. Degrees of arc, south is negative

Easternmost Longitude	ELON	R( )		real	Easternmost longitude of data coverage contained in the file indicated by the FILE subfield. Degrees of arc, west is negative
CRC	CRCS	A( )		hex	The Cyclic Redundancy Checksum for the file indicated by the FILE subfield (see 3.4)
Comment	COMT	A( )		bt	A string of characters

table 7.11

7.4.2 Catalogue cross reference record structure

```
Catalogue Cross Reference record
    |
    |--0001 (1) ISO/IEC 8211 Record Identifier
        |
        |--<R>--CATX (*5)  - Catalogue Cross Reference field
```

Field Tag: CATX	Field Name: Catalogue Cross Reference

Subfield name	Label	Format ASCII	Format Bin	Dom	Subfield content and specification
Record name	*RCNM	A(2)	b11	an	"CR" {60}
Record identification number	RCID	I(10)	b14	int	Range: 1 to 2^{32}-2
Name 1	NAM1	A(12)	B(40)	an	Foreign pointer (see 2.2)
Name 2	NAM2	A(12)	B(40)	an	Foreign pointer (see 2.2)
Comment	COMT	A( )		bt	A string of characters

table 7.12

7.5 Data dictionary records

7.5.1 Data dictionary definition record structure

```
Data Dictionary Definition record
    |
    |--0001 (1) ISO/IEC 8211 Record Identifier
        |
        |--DDDF (10)     - Data Dictionary Definition field
            |
            |--DDDR (*2)     - Data Dictionary Definition Reference field
```

7.5.1.1 Data dictionary definition field structure

Field Tag: DDDF	Field Name: Data Dictionary Definition

Subfield name	Label	Format ASCII	Format Bin	Dom	Subfield content and specification
Record name	RCNM	A(2)	b11	an	"ID" {70}
Record identification number	RCID	I(10)	b14	int	Range: 1 to 2^{32}-2
Object or attribute	OORA	A(1)	b11	an	"A" {1} The content of OAAC/OACO is an attribute "O" {2} The content of OAAC/OACO is an object
Object or attribute acronym	OAAC	A(6)		bt	A string containing an object or attribute acronym
Object or attribute label/code	OACO	I(5)	b12	int	Object or attribute label/code 1 to 8192 (IHO Object Catalogue) 8193 to 16387 (Reserved) 16388 to 65534 (General use)
Object or attribute long label	OALL	A( )		bt	A string indicating the long label of the object or attribute
Type of object or attribute	OATY	A(1)	b11	an	"M" {1} Meta object "$" {2} Cartographic object "G" {3} Geo object "C" {4} Collection object "F" {5} Feature attribute "N" {6} Feature national attribute "S" {7} Spatial attribute
Definition	DEFN	A( )		bt	A string providing a definition of the object or attribute
Authorizing agency	AUTH	A(2)	b12	an	Agency code (see IHO Object Catalogue)
Comment	COMT	A( )		bt	A string of characters

table 7.13

7.5.1.2 Data dictionary definition reference field structure

Field Tag: DDDR	Field Name: Data Dictionary Definition Reference

Subfield name	Label	Format ASCII	Format Bin	Dom	Subfield content and specification
Reference type	*RFTP	A(2)	b11	an	"I1" {1} INT 1 International chart 1, Symbols, Abbreviations, Terms used on charts "M4" {2} M-4 Chart specifications of the IHO and Regulations of the IHO for international (INT) charts
Reference value	RFVL	A( )		bt	A string containg the reference value of the type specified in the RFTP subfield

table 7.14

7.5.2 Data dictionary domain record structure

```
Data Dictionary Domain record
   |
   |--0001 (1)    - ISO/IEC 8211 Record Identifier
      |
      |--DDDI (8)   - Data Dictionary Domain Identifier field
         |
         |--<R>--DDOM (5)  - Data Dictionary Domain field
                  |
                  |--DDRF (*2)- Data Dictionary Domain Reference field
```

7.5.2.1 Data dictionary domain identifier field structure

Field Tag: DDDI	Field Name: Data Dictionary Domain Identifier

Subfield name	Label	Format ASCII	Format Bin	Dom	Subfield content and specification
Record name	RCNM	A(2)	b11	an	"IO" {80}
Record identification number	RCID	I(10)	b14	int	Range: 1 to 2^{32}-2
Attribute label/code	ATLB	I(5)	b12	int	A valid attribute label/code
Attribute domain type	ATDO	A(1)	b11	an	"E" {1} Enumerated "L" {2} List of enumerated values "F" {3} Float "I" {4} Integer "A" {5} Code string in ASCII characters "S" (6) Free text format
Attribute domain value measurement unit	ADMU	A( )		bt	A string indicating the units of measurement for values in the attribute domain
Attribute domain format	ADFT	A( )		bt	A string containing an attribute format description
Authorizing agency	AUTH	A(2)	b12	an	Agency code (see IHO Object Catalogue)
Comment	COMT	A( )		bt	A string of characters

table 7.15

7.5.2.2 Data dictionary domain field structure

Field Tag: DDOM	Field Name: Data Dictionary Domain

Subfield name	Label	Format ASCII	Format Bin	Dom	Subfield content and specification
Range or value	RAVA	A(1)	b11	an	"M" {1} DVAL contains the maximum value "N" {2} DVAL contains the minimum value "V" {3} DVAL contains a specific single value from the domain of ATDO
Domain value	DVAL	A( )		bt	A string containing a value specified by the RAVA and ATDO subfields
Domain value short description	DVSD	A( )		bt	A string contaning the short description of the domain value

Domain value definition	DEFN	A( )		bt	A string containing the definition of the domain value
Authorizing agency	AUTH	A(2)	b12	an	Agency code (see IHO Object Catalogue)

table 7.16

7.5.2.3 Data dictionary domain reference field structure

Field Tag: DDRF	Field Name: Data Dictionary Domain Reference

Subfield name	Label	Format ASCII	Bin	Dom	Subfield content and specification
Reference type	*RFTP	A(2)	b11	an	"I1" {1} INT 1 International chart 1, Symbols, Abbreviations, Terms used on charts "M4" {2} M-4 Chart specifications of the IHO and Regulations of the IHO for international (INT) charts
Reference value	RFVL	A( )		bt	A string containing the reference value of the type specified in the RFTP subfield

table 7.17

7.5.3 Data dictionary schema record structure

```
Data Dictionary Schema record
   |
   |--0001 (1)    - ISO/IEC 8211 Record Identifier
      |
      |--DDSI (3)   - Data Dictionary Schema Identifier field
         |
         |--DDSC (*3)- Data Dictionary Schema field
```

7.5.3.1 Data dictionary schema identifier field structure

Field Tag: DDSI	Field Name: Data Dictionary Schema Identifier

Subfield name	Label	Format ASCII	Bin	Dom	Subfield content and specification
Record name	RCNM	A(2)	b11	an	"IS" {90}
Record identification number	RCID	I(10)	b14	int	Range: 1 to 2^{32}-2
Object label/code	OBLB	I(5)	b12	int	A valid object label/code

table 7.18

7.5.3.2 Data dictionary schema field structure

Field Tag: DDSC	Field Name: Data Dictionary Schema

Subfield name	Label	Format ASCII	Format Bin	Dom	Subfield content and specification
Attribute label/code	*ATLB	I(5)	b12	int	A valid attribute label/code
Attribute set	ASET	A(1)	b11	an	"A" {1} Attribute set A "B" {2} Attribute set B "C" {3} Attribute set C
Authorizing agency	AUTH	A(2)	b12	an	Agency code (see IHO Object Catalogue)

table 7.19

7.6 Feature record structure

```
Feature record
    |
    |--0001 (1)    - ISO/IEC 8211 Record Identifier
        |
        |--FRID (7)   - Feature record identifier field
            |
            |--FOID (3)             - Feature object identifier field
            |
            |--<R>--ATTF (*2)  - Feature record attribute field
            |
            |--<R>--NATF (*2)  - Feature record national attribute field
            |
            |--FFPC (3)             - Feature record to feature object pointer control field
            |
            |--<R>--FFPT (*3)  - Feature record to feature object pointer field
            |
            |--FSPC (3)             - Feature record to spatial record pointer control field
            |
            |--<R>--FSPT (*4)  - Feature record to spatial record pointer field
```

7.6.1 Feature record identifier field structure

Field Tag: FRID	Field Name: Feature Record Identifier

Subfield name	Label	Format ASCII	Format Bin	Dom	Subfield content and specification
Record name	RCNM	A(2)	b11	an	"FE" {100}
Record identification number	RCID	I(10)	b14	int	Range: 1 to 2^{32}-2
Object geometric primitive	PRIM	A(1)	b11	an	"P" {1} Point "L" {2} Line "A" {3} Area "N" {255} Object does not directly reference any spatial objects (see 4.2.1)
Group	GRUP	I(3)	b11	int	Range: 1 to 254, 255 - No group (binary) (see Appendix B - Product Specifications)

Object label/code	OBJL	I(5)	b12	int	A valid object label/code
Record version	RVER	I(3)	b12	int	RVER contains the serial number of the record edition (see 8.4.2.1)
Record update instruction	RUIN	A(1)	b11	an	"I" {1} Insert "D" {2} Delete "M" {3} Modify (see 8.4.2.2)

table 7.20

7.6.2 Feature object identifier field structure

Field Tag: FOID	Field Name: Feature Object Identifier

Subfield name	Label	Format ASCII	Bin	Dom	Subfield content and specification
Producing agency	AGEN	A(2)	b12	an	Agency code (see 4.3)
Feature identification number	FIDN	I(10)	b14	int	Range: 1 to 2^{32}-2 (see 4.3.2)
Feature identification subdivision	FIDS	I(5)	b12	int	Range: 1 to 2^{16}-2 (see 4.3.2)

table 7.21

7.6.3 Feature record attribute field structure

Field Tag: ATTF	Field Name: Feature record attribute

Subfield name	Label	Format ASCII	Bin	Dom	Subfield content and specification
Attribute label/code	*ATTL	I(5)	b12	int	A valid attribute label/code
Attribute value	ATVL	A( )		gt	A string containing a valid value for the domain specified by the attribute label/code in ATTL

table 7.22

7.6.4 Feature record national attribute field structure

Field Tag: NATF	Field Name: Feature record national attribute

Subfield name	Label	Format ASCII	Bin	Dom	Subfield content and specification
Attribute label/code	*ATTL	I(5)	b12	int	A valid national attribute label/code
Attribute value	ATVL	A( )		gt	A string containing a valid value for the domain specified by the attribute label/code in ATTL

table 7.23

7.6.5 Feature record to feature object pointer control field structure

Field Tag: FFPC [Upd]	Field Name: Feature Record to Feature Object Pointer Control

Subfield name	Label	Format ASCII	Bin	Dom	Subfield content and specification
Feature object pointer update instruction	FFUI	A(1)	b11	an	"I" {1} Insert "D" {2} Delete "M" {3} Modify (see 8.4.2.3)
Feature object pointer index	FFIX	I( )	b12	int	Index (position) of the adressed record pointer within the FFPT field(s) of the target record (see 8.4.2.3)
Number of feature object pointers	NFPT	I( )	b12	int	Number of record pointers in the FFPT field(s) of the update record (see 8.4.2.3)

table 7.24

7.6.6 Feature record to feature object pointer field structure

Field Tag: FFPT	Field Name: Feature Record to Feature Object Pointer

Subfield name	Label	Format ASCII	Bin	Dom	Subfield content and specification
Long Name	*LNAM	A(17)	B(64)	an	Foreign pointer (see 4.3)
Relationship indicator	RIND	A( )	b11	an	"M" {1} Master "S" {2} Slave "P" {3} Peer Other values may be defined by the relevant product specification (see 6.2 and 6.3)
Comment	COMT	A( )		bt	A string of characters

table 7.25

7.6.7 Feature record to spatial record pointer control field structure

Field Tag: FSPC [Upd]	Field Name: Feature Record to Spatial Record Pointer Control

Subfield name	Label	Format ASCII	Bin	Dom	Subfield content and specification
Feature to spatial record pointer update instruction	FSUI	A(1)	b11	an	"I" {1} Insert "D" {2} Delete "M" {3} Modify (see 8.4.2.4)
Feature to spatial record pointer index	FSIX	I( )	b12	int	Index (position) of the adressed record pointer within the FSPT field(s) of the target record (see 8.4.2.4)
Number of feature to spatial record pointers	NSPT	I( )	b12	int	Number of record pointers in the FSPT field(s) of the update record (see 8.4.2.4)

table 7.26

7.6.8 Feature record to spatial record pointer field structure

Field Tag: FSPT	Field Name: Feature Record to Spatial Record Pointer

Subfield name	Label	Format ASCII	Format Bin	Dom	Subfield content and specification
Name	*NAME	A(12)	B(40)	an	Foreign pointer (see 2.2)
Orientation	ORNT	A(1)	b11	an	"F" {1} Forward "R" {2} Reverse "N" {255} NULL
Usage indicator	USAG	A(1)	b11	an	"E" {1} Exterior "I" {2} Interior "C" {3} Exterior boundary truncated by the data limit "N" {255} NULL
Masking indicator	MASK	A(1)	b11	an	"M" {1} Mask "S" {2} Show "N" {255} NULL

table 7.27

7.7 Spatial record structure

7.7.1 Vector record structure

```
Vector record
 |
 |--0001 (1)    - ISO/IEC 8211 Record Identifier
     |
     |--VRID (4)   - Vector Record Identifier field
         |
         |--<R>--ATTV (*2)  - Vector Record Attribute field
         |
         |--VRPC (3)           - Vector Record Pointer Control field
         |
         |--<R>--VRPT (*5)  - Vector Record Pointer field
         |
         |--SGCC (3)           - Coordinate control field
         |
         |    alternate coordinate representations
         |
         *--<R>--SG2D (*2) - 2-D Coordinate field
         |
         *--<R>--SG3D (*3) - 3-D coordinate (Sounding Array) field
         |
         *--<R>--ARCC (5)  - Arc/Curve definitions field
                     |
                     |    alternate arc/curve definitions
                     |
                     *--<R>--AR2D (3*2)    - Arc coordinates field
                     |
                     *--<R>--EL2D (5*2)    - Ellipse coordinates field
                     |
                     *--<R>--CT2D (*2)      - Curve coordinates field
```

7.7.1.1 Vector record identifier field structure

Field Tag: VRID	Field Name: Vector Record Identifier

Subfield name	Label	Format ASCII	Format Bin	Dom	Subfield content and specification
Record name	RCNM	A(2)	b11	an	"VI" {110} Isolated node "VC" {120} Connected node "VE" {130} Edge "VF" {140} Face
Record identification number	RCID	I(10)	b14	int	Range: 1 to 2^{32}-2
Record version	RVER	I(3)	b12	int	RVER contains the serial number of the record edition (see 8.4.3.1)
Record update instruction	RUIN	A(1)	b11	an	"I" {1} Insert "D" {2} Delete "M" {3} Modify (see 8.4.3.2)

table 7.28

7.7.1.2 Vector record attribute field structure

Field Tag: ATTV	Field Name: Vector Record Attribute

Subfield name	Label	Format ASCII	Format Bin	Dom	Subfield content and specification
Attribute label/code	*ATTL	I(5)	b12	int	A valid attribute label/code
Attribute value	ATVL	A( )		bt	A string containing a valid value for the domain specified by the attribute label/code in ATTL

table 7.29

7.7.1.3 Vector record pointer control field structure

Field Tag: VRPC [Upd]	Field Name: Vector Record Pointer Control

Subfield name	Label	Format ASCII	Format Bin	Dom	Subfield content and specification
Vector record pointer update instruction	VPUI	A(1)	b11	an	"I" {1} Insert "D" {2} Delete "M" {3} Modify (see 8.4.3.2.b)
Vector record pointer index	VPIX	I( )	b12	int	Index (position) of the adressed vector record pointer within the VRPT field(s) of the target record (see 8.4.3.2.b)
Number of vector record pointers	NVPT	I( )	b12	int	Number of vector record pointers in the VRPT field(s) of the update record (see 8.4.3.2.b)

table 7.30

7.7.1.4 Vector record pointer field structure

Field Tag: VRPT	Field Name: Vector Record Pointer

Subfield name	Label	Format ASCII	Format Bin	Dom	Subfield content and specification
Name	*NAME	A(12)	B(40)	an	Foreign pointer (see 2.2)
Orientation	ORNT	A(1)	b11	an	"F" {1} Forward "R" {2} Reverse "N" {255} NULL (see 5.1.3)
Usage indicator	USAG	A(1)	b11	an	"E" {1} Exterior "I" {2} Interior "C" {3} Exterior boundary truncated by the data limit "N" {255} NULL (see 5.1.3)
Topology indicator	TOPI	A(1)	b11	an	"B" {1} Beginning node "E" {2} End node "S" {3} Left face "D" {4} Right face "F" {5} Containing face "N" {255} NULL (see 5.1.3)
Masking indicator	MASK	A(1)	b11	an	"M" {1} Mask "S" {2} Show "N" {255} NULL (see 5.1.3)

table 7.31

7.7.1.5 Coordinate control field structure

Field Tag: SGCC [Upd]	Field Name: Coordinate control

Subfield name	Label	Format ASCII	Format Bin	Dom	Subfield content and specification
Coordinate update instruction	CCUI	A(1)	b11	an	"I" {1} Insert "D" {2} Delete "M" {3} Modify (see 8.4.3.3)
Coordinate index	CCIX	I( )	b12	int	Index (position) of the adressed coordinate within the coordinate field(s) of the target record (see 8.4.3.3)
Number of coordinates	CCNC	I( )	b12	int	Number of coordinates in the coordinate field(s) of the update record (see 8.4.3.3)

table 7.32

7.7.1.6 2-D Coordinate field structure

Field Tag: SG2D	Field Name: 2-D Coordinate

Subfield name	Label	Format ASCII	Format Bin	Dom	Subfield content and specification
Coordinate in Y axis	*YCOO	R( )	b24	real	Y coordinate. Format is specified in Appendix B - Product Specification
Coordinate in X axis	XCOO	R( )	b24	real	X coordinate. Format is specified in Appendix B - Product Specification

table 7.33

7.7.1.7 3-D Coordinate field structure

Field Tag: SG3D	Field Name: 3-D Coordinate (Sounding Array)

Subfield name	Label	Format ASCII	Format Bin	Dom	Subfield content and specification
Coordinate in Y axis	*YCOO	R( )	b24	real	Y coordinate. Format is specified in Appendix B - Product Specifications
Coordinate in X axis	XCOO	R( )	b24	real	X coordinate. Format is specified in Appendix B - Product Specifications
3-D (sounding) value	VE3D	R( )	b24	real	Value of third dimension. Content and format are specified in Appendix B - Product Specifications

table 7.34

7.7.1.8 Arc/Curve definition field structure

Field Tag: ARCC	Field Name: Arc/Curve definition

Subfield name	Label	Format ASCII	Format Bin	Dom	Subfield content and specification
Arc/Curve type	ATYP	A(1)	b11	an	"C" {1} Arc 3 point centre "E" {2} Elliptical arc "U" {3} Uniform Bspline "B" {4} Piecewise bezier "N" {5} Non-uniform rational B-spline (see 5.1.4.4)
Construction surface	SURF	A(1)	b11	an	"E" {1} Ellipsoidal Object must be reconstructed prior to projection onto a 2-D surface "P" {2} Planar Object must be reconstructed after projection onto a 2-D surface, regardless of projection used
Curve order	ORDR	I(1)	b11	int	Value of the largest exponent of the polynomial equation Range: 1 to 9

Interpolated point resolution	RESO	R( )	b14 *)	real	Spacing along line path between interpolated points. Value in map units (millimeters)
Floating point multiplication factor	FPMF	I( )	b14	int	Floating point to integer multiplication factor for interpolated point resolution value (see 2.6)

table 7.35

*)use FPMF subfield to convert integer (b14) to floating point (see clause 2.6).

7.7.1.9 Arc coordinates field structure

Field Tag: AR2D	Field Name: Arc coordinate

Subfield name	Label	Format ASCII	Bin	Dom	Subfield content and specification
Start point	STPT	@			ISO/IEC 8211 Cartesian label
Centre point	CTPT	@			ISO/IEC 8211 Cartesian label
End point	ENPT	@			ISO/IEC 8211 Cartesian label
Coordinate in Y axis	*YCOO	R( )	b24	real	Y coordinate. Format is specified in Appendix B - Product Specifications
Coordinate in X axis	XCOO	R( )	b24	real	X coordinate. Format is specified in Appendix B - Product Specifications

table 7.36

7.7.1.10 Ellipse coordinates field structure

Field Tag: EL2D	Field Name: Ellipse coordinates

Subfield name	Label	Format ASCII	Bin	Dom	Subfield content and specification
Start point	STPT	@			ISO/IEC 8211 Cartesian label
Centre point	CTPT	@			ISO/IEC 8211 Cartesian label
End point	ENPT	@			ISO/IEC 8211 Cartesian label
Conjugate diameter point major axis	CDPM	@			ISO/IEC 8211 Cartesian label
Conjugate diameter point minor axis	CDPR	@			ISO/IEC 8211 Cartesian label
Coordinate in Y axis	*YCOO	R( )	b24	real	Y coordinate. Format is specified in Appendix B - Product Specifications
Coordinate in X axis	XCOO	R( )	b24	real	X coordinate. Format is specified in Appendix B - Product Specifications

table 7.37

7.7.1.11 Curve coordinates field structure

Field Tag: CT2D	Field Name: Curve Coordinates

Subfield name	Label	Format ASCII	Format Bin	Dom	Subfield content and specification
Coordinate in Y axis	*YCOO	R( )	b24	real	Y coordinate. Format is specified in Appendix B - Product Specifications
Coordinate in X axis	XCOO	R( )	b24	real	X coordinate. Format is specified in Appendix B - Product Specifications

table 7.38

7.7.2 Raster record structure

To be defined.

7.7.3 Matrix record structure

To be defined.

8. Updating

8.1 General

This chapter defines a mechanism for the updating of S-57 conforming data. The mechanism allows for the updating of individual constructs (records, fields and subfields) within the data. By using this mechanism, previousely exchanged data can be brought up to date without the need for reissuing a complete new set of data.
The update mechanism reflects the S-57 data structure. Therefore, the highest level construct which can be updated by using this mechanism is a record (see also clause 1.1). Procedures for the updating of complete files must be defined by the relevant product specification.

To facilitate updating, special fields have been added to the feature and spatial records (see clause 7.6 and 7.7). These fields are only used for updating. Currently only spatial records of type vector are defined by this Standard. Therefore, the update mechanism for spatial records has only been developed for vector records.

8.2 Update data flow

The updating mechanism is based on an exchange model which is shown in figure 8.1.

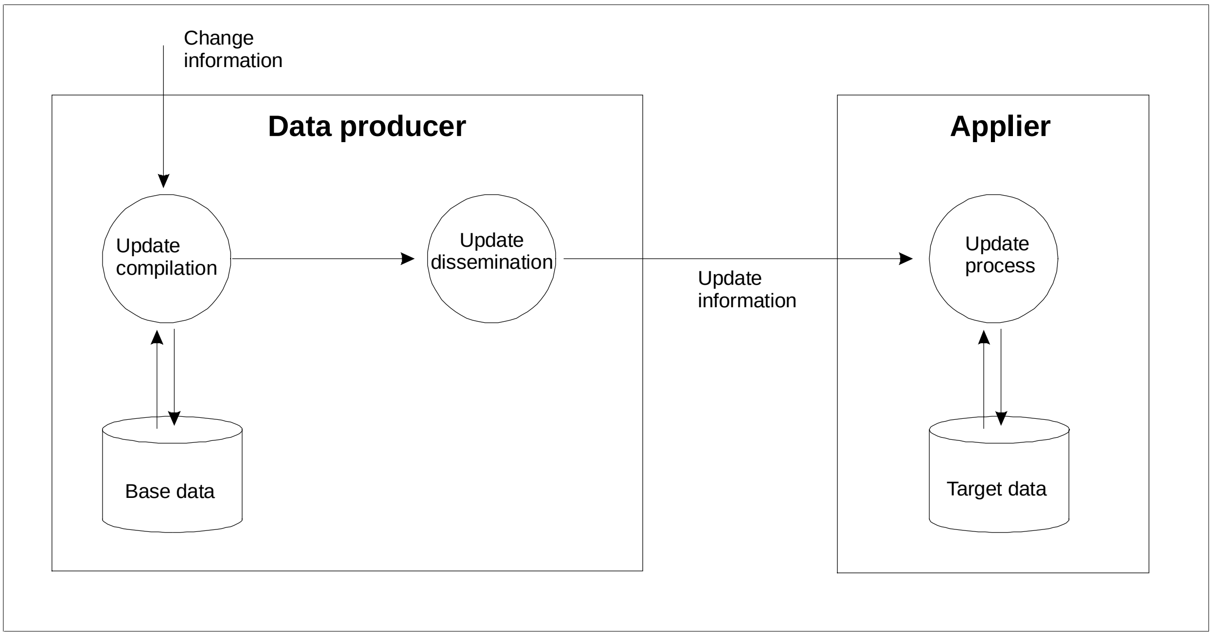

figure 8.1

This model illustrates the update data flow from data producer to applier. The organization of the various processes is application dependent. In general the data flow for the update mechanism can be described as follows:

Updates to the base data are compiled based on “change” information received by the data producer. The resulting update information which comprises one or more update records is disseminated. On receipt the applier applies the update information to the target data in what is called the update process. The application of a single update record as part of the update process is called an update operation.

8.3 Identity of constructs within the update mechanism

8.3.1 General

To facilitate updating, records, fields and subfields must be uniquely identifiable. Clauses 8.3.2 to 8.3.5 describe how the various data structure constructs are identified within the update mechanism.

In order for the applier to perform the update operation the original identity of data structure constructs in the base data must be preserved in the target data.

8.3.2 Record identity within the update mechanism

Records are identified following the rules given in clause 2.2. The NAME of a record must be preserved on creation and must be re-used in each subsequent update operation refering to that record.

8.3.3 Attribute identity within the update mechanism

Attributes are identified by a combination of the NAME of the record in which they are contained and the unique attribute label/code defined in the IHO Object Catalogue (see clause 4.4). Since attributes must not repeat within a record, attribute labels/codes are used as unique identifiers within the scope of a record.

8.3.4 Pointer identity within the update mechanism

Individual pointers *) within the pointer fields (FFPT, FSPT and VRPT) are identified by a combination of the NAME of the record in which they are contained and an index. The index identifies the position of the pointer within the pointer fields. For pointer fields which carry only one pointer the index is set to one. For pointer fields which carry more than one pointer the index will be equal to or greater than one.
If a feature or vector record contains repeating pointer fields (e.g. two or more FSPT fields within one feature record) the index is continued from one field to the other (i.e. if the first pointer field contains n pointers, the index of the first pointer of the second field becomes n+1).

*) In the context of the update mechanism, the pointer index always refers to the pointer and its associated subfields (e.g. for the FFPT the index refers to a combination of the LNAM and RIND subfield).

8.3.5 Coordinate identity within the update mechanism

Individual coordinates are identified by a combination of the NAME of the vector record in which they are contained and a coordinate index *). The index identifies the position of the coordinate within the coordinate fields of the vector record (see clause 5.1.4). For vector records which may carry only one coordinate (e.g. connected nodes or isolated nodes) the index is set to one. For vector records which carry more than one coordinate (e.g. edges and sounding arrays) the index will be equal to or greater than one.
If a vector record contains repeating coordinate fields (e.g. two or more SG2D fields within one edge vector record) the index is continued from one field to the other (i.e. if the first coordinate field contains n coordinates, the index of the first coordinate of the second field becomes n+1).
During the conversion between base data and target data coordinates must not be added to or removed from the coordinate fields since this will change the indices of the coordinates.

*) In the context of the update mechanism, a coordinate is a coordinate pair (YCOO, XCOO) for the [SG2D], [AR2D], [EL2D] and [CT2D] fields and a triplet (YCOO, XCOO, VE3D) for the [SG3D] field.

8.4 Update record

8.4.1 General

Updating of data is achieved by applying update records to the target data. An update record is a generic term for a feature or vector record that contains a delete or modify indicator in its “Record Update instruction” [RUIN] subfield. An update record may also contain additional fields. A feature record may contain the following additional (update instruction) fields:

FFPC - Feature Record to Feature Record Pointer Control field
FSPC - Feature Record to Spatial Record Pointer Control field

A vector record may contain the following additional (update instruction) fields:

VRPC - Vector Record Pointer Control field
SGCC - Coordinate Control field

The structure of the update fields is given in the clauses 7.6 and 7.7.1. An explanation of their use is given in the clauses 8.4.2 and 8.4.3.

8.4.2 Constructs for updating the feature record

8.4.2.1 Record version subfield — use for feature record

The “Record Version” [RVER] subfield carries the version number for the record in which it is contained. The version number of the update record must be one higher than the version number of the target record to which the update record applies. On completion of the update operation the version number of the target record must be made equal to the version number of the update record applied to it.

8.4.2.2 Record update instruction subfield — use for feature record

The “Record Update Instruction” [RUIN] subfield defines which update operation must be performed on the target record. This subfield may carry one of the following values:

I {1} INSERT - Feature record must be inserted. The RVER subfield must contain “1".

D {2} DELETE - Feature record must be deleted. This record must not contain further fields; only FRID is used.

M {3} MODIFY - Feature record must be modified. This record must contain further fields conveying the details of this update message. The “M” value indicates that one or more fields of the target record will be affected by the update operation. The rules for modifying these fields are given below.

a. Modification of the ATTF and NATF fields

When an update record contains an ATTF and/or NATF field it must be interpreted in the following way. If the attribute does not occur in the target record, the attribute must be treated as an insertion. If an attribute already exist in the target record its value must be replaced by the value of the corresponding attribute held in the update record. An attribute is removed from an object by sending the attribute in the update record with its value set to the delete character. Table 8.1 defines the delete character for each lexical level.

Lexical level	Delete
Level 0	(7/15)
Level 1	(7/15)
Level 2	(0/0) (7/15)

table 8.1

The lexical level used for the ATTF and NATF fields in the update record must correspond with the lexical level used for the ATTF and NATF fields in the target record (see clause 2.4).

b. Modification of the FFPT and FSPT fields

The modification of the pointer fields (FFPT and FSPT) is controlled by the pointer control fields. The FFPC and FSPC pointer control fields are specified in clause 8.4.2.3 and 8.4.2.4 respectively.

8.4.2.3 Feature record to feature object pointer control field

The “Feature Record to Feature Object Pointer Control” [FFPC] field controls the updating of the “Feature Record to Feature Object Pointer” [FFPT] field(s). The FFPC field contains three subfields:

FFUI - Feature Object Pointer Update Instruction subfield
FFIX - Feature Object Pointer Index subfield
NFPT - Number of Feature Object Pointers subfield

The “Feature Object Pointer Update Instruction” [FFUI] subfield may carry one of the following values:

I {1} INSERT - Feature record to feature object pointer(s) encoded in the FFPT field(s) of the update record must be inserted in the FFPT field(s) of the target record. The insertion must start *at the position one before* the index specified by the FFIX subfield. The number of pointers to be inserted is given in the NFPT subfield.

D {2} DELETE - Feature record to feature object pointer(s) must be deleted from the FFPT field(s) of the target record. The deletion must start at the index specified in the FFIX subfield. The number of pointers to be removed is given in the NFPT subfield.

M {3} MODIFY - Feature record to feature object pointer(s) encoded in the FFPT field(s) of the update record must replace the adressed pointer(s) in FFPT field(s) of the target record. The replacement must start at the index given in the FFIX subfield. The number of pointers to be replaced is given in the NFPT subfield.

The "Feature Object Pointer Index" [FFIX] subfield gives the position of the addressed “feature record to feature object pointer” within the FFPT field(s) of the target record (see clause 8.3.4).

The "Number of Feature Object Pointers" [NFPT] subfield gives the number of “feature record to feature object pointers” in the FFPT field(s) in the update record.

The FFPC field is not allowed to repeat within an update record. An update to non-consecutive pointers (e.g. the first and the last pointer in a FFPT field containing many pointers) can be handled by multiple update records or by replacement of all feature record to feature object pointers in the target record.

8.4.2.4 Feature record to spatial record pointer control field

The “Feature Record to Spatial Record Pointer Control” [FSPC] field controls the updating of the “Feature Record to Spatial Record Pointer” [FSPT] field(s). The FSPC field contains three subfields:

FSUI - Feature to Spatial Record Pointer Update Instruction subfield
FSIX - Feature to Spatial Record Pointer Index subfield
NSPT - Number of Feature to Spatial Record Pointers subfield

The "Feature to Spatial Record Pointer Update Instruction" [FSUI] subfield may carry one of the following values:

I {1} INSERT - Feature to spatial record pointer(s) encoded in the FSPT field(s) of the update record must be inserted in the FSPT field(s) of the target record. The insertion must start *at the position one before* the index specified by the FSIX subfield. The number of pointers to be inserted is given in the NSPT subfield.

D {2} DELETE - Feature to spatial record pointer(s) must be deleted from the FSPT field(s) of the target record. The deletion must start at the index specified in the FSIX subfield. The number of pointers to be removed is given in the NSPT subfield.

M {3} MODIFY - Feature to spatial record pointer(s) encoded in the FSPT field(s) of the update record must replace the addressed pointer(s) in FSPT field(s) of the target record. The replacement must start at the index given in the FSIX subfield. The number of pointers to be replaced is given in the NSPT subfield.

The "Feature to Spatial Record Pointer Index" [FSIX] subfield gives the position of the addressed “Feature to Spatial Record Pointer” within the FSPT field(s) of the target record (see clause 8.3.4).

The "Number of Feature to Spatial Record Pointers" [NSPT] subfield gives the number of “Feature to Spatial Record Pointers” in the FSPT field(s) in the update record.

The FSPC field is not allowed to repeat within an update record. An update to non-consecutive pointers (e.g. the first and the last pointer in a FSPT field containing many pointers) can be handled by multiple update records or by replacement of all feature to spatial record pointers in the target record.

8.4.3 Constructs for updating the vector record

8.4.3.1 Record version subfield — use for vector record

The “Record Version” [RVER] subfield carries the version number for the record in which it is contained. The version number of the update record must be one higher than the version number of the target record to which the update record applies. On completion of the update operation the version number of the target record must be made equal to the version number of the update record applied to it.

8.4.3.2 Record update instruction subfield — use for vector record

The “Record Update Instruction” [RUIN] subfield defines which update operation must be performed on the target record. This subfield may carry one of the following values:

I {1} INSERT - Vector record must be inserted. The RVER subfield must contain “1".

D {2} DELETE - Vector record must be deleted. This record must not contain further fields; only VRID is used.

M {3} MODIFY - Vector record must be modified. This record must contain further fields conveying the details of this update message. The “M” value indicates that one or more fields of the target record will be affected by the update operation. The rules for modifying these fields are given below.

a. Modification of the ATTV field

When an update record contains an ATTV field it must be interpreted in the following way. If the attribute does not occur in the target record, the attribute must be treated as an insertion. If an attribute already exist in the target record its value must be replaced by the value of the corresponding attribute held in the update record. An attribute is removed from an object by sending the attribute in the update record with its value set to the delete character (7/15).

b. Modification of the VRPT field

The “Vector Record Pointer Control” [VRPC] field controls the updating of the “Vector Record Pointer” [VRPT] field(s). The VRPC field contains three subfields:

VPUI - Vector Record Pointer Update Instruction subfield
VPIX - Vector Record Pointer Index subfield
NVPT - Number of Vector Record Pointers subfield

The "Vector Record Pointer Update Instruction" [VPUI] subfield may carry one of the following values:

I {1} INSERT - Vector record pointer(s) encoded in the VRPT field(s) of the update record must be inserted in the VRPT field(s) of the target record. The insertion must start *at the position one before* the index specified by the VPIX subfield. The number of pointers to be inserted is given in the NVPT subfield.

D {2} DELETE - Vector record pointer(s) must be deleted from the VRPT field(s) of the target record. The deletion must start at the index specified in the VPIX subfield. The number of pointers to be removed is given in the NVPT subfield.

M {3} MODIFY - Vector record pointer(s) encoded in the VRPT field(s) of the update record must replace the addressed pointer(s) in VRPT field(s) of the target record. The replacement must start at the index given in the VPIX subfield. The number of pointers to be replaced is given in the NVPT subfield.

The "Vector Record Pointer Index" [VPIX] subfield gives the position of the addressed “Vector Record Pointer” within the VRPT field(s) of the target record (see clause 8.3.4).

The "Number of Vector Record Pointers" [NVPT] subfield gives the number of “Vector Record Pointers” in the VRPT field(s) in the update record.

The VRPC field is not allowed to repeat within an update record. An update to non-consecutive pointers (e.g. the first and the last pointer in a VRPT field containing many pointers) can be handled by multiple update records or by replacement of all spatial record pointers in the target record.

c. Modification of the SG2D, SG3D, AR2D and EL2D fields

The modification of coordinate fields (see clause 5.1.4) is controlled by the “Coordinate Control” [SGCC] field. The SGCC field is specified in clause 8.4.3.3.

8.4.3.3 Coordinate control field

The “Coordinate Control” [SGCC] field controls the updating of coordinates fields of a vector record (chapter 5). The SGCC field contains three subfields:

CCUI - Coordinate Update Instruction subfield
CCIX - Coordinate Index subfield
CCNC - Number of Coordinates subfield

The “Coordinate Update Instruction” [CCUI] subfield may carry one of the following values:

I {1} INSERT - Coordinate(s) encoded in the coordinate field(s) of the update record must be inserted in the coordinate field(s) of the target record. The insertion must start *at the position one before* the index specified by the CCIX subfield. The number of coordinates to be inserted is given in the CCNC subfield.

D {2} DELETE - Coordinate(s) must be deleted from the coordinate field(s) of the target record. The deletion must start at the index specified in the CCIX subfield. The number of coordinates to be removed is given in the CCNC subfield.

M {3} MODIFY - Coordinate(s) encoded in the coordinate field(s) of the update record must replace the addressed coordinate(s) in coordinate field(s) of the target record. The replacement must start at the index given in the CCIX subfield. The number of coordinates to be replaced is given in the CCNC subfield.

The “Coordinate Index” [CCIX] subfield gives the position of the addressed coordinate within the coordinate field(s) of the target record (see clause 8.3.5).

The “Number of Coordinates” [CCNC] subfield gives the number of coordinates in the coordinate field(s) in the update record.

The SGCC field is not allowed to repeat within an update record. An update to non-consecutive coordinates (e.g. the first and the last coordinate in a SG2D field which contains many coordinates) can be handled by multiple update records or by replacement of all coordinates in the target record.

In the case of inserting one or more coordinates into a target record representing a straight line (i.e. an edge which only references its connecting nodes, see clause 5.1.4.4), an update record must not contain a SGCC field and the coordinate fields of the update record must be added to the target record without further modification.

Page intentionally left blank

Important notice

All “Clarifications” in the latest edition of the Maintenance Document must be taken into account before making use of this document.

A. ISO/IEC 8211 summary and examples

S-57 uses ISO/IEC 8211:1994 as its encapsulation standard. ISO/IEC 8211 specifies an interchange format to facilitate the exchange of data between computer systems. The intention of this annex is to give a short overview of ISO/IEC 8211 by explaining those parts of the encapsulation structure which are of relevance to S-57. The explanation is illustrated by a simple example comprising one buoy at a given location (i.e. a geo feature record and a vector spatial record).

This annex does not replace ISO/IEC 8211 nor does it give a comprehensive overview of ISO/IEC 8211. The reader is referred to ISO/IEC 8211:1994 for a complete description and explanation.

A.1 Relationship between the data structure and the encapsulation

Clause 1.1 of Part 3 of S-57 explains how the various constructs from the theoretical data model (Part 2 of S-57) are translated into the S-57 data structure. The relationship between the structure and ISO/IEC 8211 encapsulation constructs is shown below.

S-57 Data Structure	**Encapsulation**
Record..	Logical Record (LR) containing a group of fields
Field..	Field
Subfield...	Subfield

The grouping of fields into a LR is defined in the S-57 data structure by means of the tree structure diagrams. Each tree structure diagram in chapter 7 of Part 3 of S-57defines a S-57 record.

The S-57 data structure does not specify a file structure. The grouping of records into files is product specific and, therefore, defined by the relevant product specification.

A.2 ISO/IEC 8211 interchange file structure

A.2.1 Logical records

ISO/IEC 8211 is a file based exchange format. The basic component of an ISO/IEC 8211 file is a logical record (LR). The first LR of an ISO/IEC 8211 file is called the "Data Descriptive Record" (DDR) and contains the description and logical structure of the actual data contained in the file. For a S-57 file, the DDR contains the information given in the tree structure diagrams and tables which are specified in chapter 7 of Part 3 of S-57. All other records in the ISO/IEC 8211 files are called "Data Records" and contain the actual data to be exchanged. The basic file structure (one DDR and one or more DR's) is shown in figure A.1.

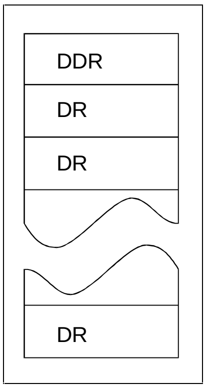

figure A.1

Each logical record (both DDR and DR) contains three basic elements:

- Leader
- Directory
- Field area

A.2.2 Leader

The leader of a logical record contains the parameters necessary to read records and to disaggregate the directory (see clause A.2.3) into its entries. In addition, the DDR leader contains a few data descriptive parameters applicable to the entire file.

A.2.2.1 DDR leader

The structure of the DDR leader is shown in table A.1.

RP	Len	Entry name	Content
0	5	Record length	number of bytes in record
5	1	Interchange level	"3"
6	1	Leader identifier	"L"
7	1	In line code extension indicator	"E"
8	1	Version number	"1"
9	1	Application indicator	SPACE
10	2	Field control length	"09"
12	5	Base address of field area	Start address of field area (number of bytes in leader and directory)
17	3	Extended character set indicator	" ! " (SPACE,!,SPACE)
20	4	Entry map	(see table A.2)

table A.1

Entry map of DDR leader

RP	Sub-entry name	Len	Content
20	Size of field length field	1	Variable 1-9 (defined by encoder)
21	Size of field position field	1	Variable 1-9 (defined by encoder)
22	Reserved	1	"0"
23	Size of field tag field	1	"4"

table A.2

A.2.2.2 DR leader

The structure of the DR leader is shown in table A.3.

RP	Len	Entry name	Content
0	5	Record length	number of bytes in record
5	1	Interchange level	SPACE
6	1	Leader identifier	"D"
7	1	In line code extension indicator	SPACE
8	1	Version number	SPACE
9	1	Application indicator	SPACE
10	2	Field control length	2 SPACEs
12	5	Base address of field area	Start address of field area (number of bytes in leader and directory)
17	3	Extended character set indicator	3 SPACEs
20	4	Entry map	(see table A.4)

table A.3

Entry map of DR leader

RP	Sub-entry name	Len	Content
20	Size of field length field	1	Variable 1-9 (defined by encoder)
21	Size of field position field	1	Variable 1-9 (defined by encoder)
22	Reserved	1	"0"
23	Size of field tag field	1	"4"

table A.4

A.2.3 Directory

The directory of a logical record contains the parameters necessary to identify and locate each field in the field area (see clause A.2.4).

The directory consists of repeated directory entries containing the field tag, field length and field position. The directory ends with the field terminator (1/14). The field positions are relative to the beginning of the field area. The position of the first field following the directory is 0. The number of bytes used for the three elements (the field entry) is defined by the entry map in the leader of the logical record.

A.2.4 Field area

The field area is different for the DDR and DR. In the first record only, the DDR, the field area contains data descriptive fields. Each data descriptive field contains information necessary to decode the user data in the field area of the DR('s).

The fact that the data description is contained in the interchange file makes it possible to exchange data without an external description. The S-57 Standard (including the relevant product specifications) does contain an external data description used for the exchange of digital hydrographic data (chapter 7 of Part 3, tree structure diagrams and tables). However, the data descriptive fields cannot be omitted from the DDR despite the existence of an external data description. The data descriptive fields of the DDR form an integral part of an ISO/IEC 8211 conforming file.

The field area of the DR contains the actual data to be transferred .

A.2.4.1 Field area of the DDR

a. Field control field

The first field of the DDR is the field control field. The field tag for the field control field is "0000". The field control field contains a list of field tag pairs. The list defines the parent/offspring binary relation of all the fields described in the DDR. The list together with the preorder traversal sequence of the field descriptions in the DDR describes a generic tree structure for the exchange file. The pairs may be placed in the list in any sequence and must be contiguous. Figure A.2 gives an example of a tree. The set of field tag pairs is HE, EA, EB, HF, HG, GC and GD.

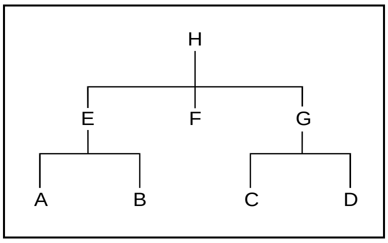

figure A.2

The structure of the field control field is shown in table A.5.

Field controls	External file title (Not used for S-57)	UT	List of field tag pairs	FT

table A.5

The field controls of the field control field are a special case of the field controls described in clause A.2.4.1.b. The nine bytes of the field controls contain "0000;&" + 3 SPACEs.

b. Data descriptive fields

The successive fields of the field area contain the data descriptive fields. The data descriptive fields are encoded in the DDR in a preorder traversal sequence. The preorder traversal sequence of the tree shown in figure A.2 is HEABFGCD.

The structure of a data descriptive field is shown in table A.6

Field controls	Field name	UT	Array descriptor	UT	Format controls	FT

table A.6

The field controls describe the level and data type of the data fields defined by the data descriptive fields. The structure of the field controls is shown in table A.7.

RP	Len	Entry name	Content
0	1	Data structure code	"1" - linear structure "2" - multi-dimensional structure
1	1	Data type code	"0" - character string "1" - implicit point (integer) "5" - binary form "6" - mixed data types
2	2	Auxiliary controls	"00"
4	2	Printable graphics	";&"
6	3	Truncated escape sequence	lexical level 0 - " " (SPACEs) lexical level 1 - "-A " (-,A,SPACE) lexical level 2 - "%/A"

table A.7

The field name contains the long description of the data fields as defined in the tree structures given in chapter 7 of Part 3 of S-57. The Array description and format controls define the inner field structure for the associated data fields. Refer to ISO/IEC 8211 for a complete description.

A.2.4.2 Field area of the DR's

The data fields in the DR's must be encoded in the preorder traversal sequence as defined in the DDR. The structure of the data fields is defined by the data descriptive fields in the DDR.

A.3 Use of ISO/IEC 8211 for S-57

ISO/IEC 8211 provides a mechanism for both numeric and binary encoding of the record length field (LR RP 0), the base address field (LR RP 12), the field length field (directory) and the field position field (directory). The numeric form must be used for S-57 conforming data.

The numeric and binary encoding of the above mentioned fields should not be confused with the binary and ASCII implementation options for the S-57 data structure. The S-57 binary and ASCII implementations refer to the data themselves and not to the ISO/IEC 8211 constructs.

Clause A.4 contains two examples of ISO/IEC 8211 encapsulations. The examples must be followed exactly by the encoder. Only the size of the field length field and the size of field position field in the entry map are variable and may be defined by the encoder (in the examples both are set to 5).

A.4 Example of an exchange file

This clause contains an example of an ISO/IEC 8211 conforming file containing S-57 data. The example file contains one geo feature record and one vector spatial record. The geo feature record contains a buoy object. The vector spatial record contains the position of that buoy. The grouping of the records in this file is arbitrary and is only used as an example. The grouping of records for a given application must be defined by a relevant product specification. The DDR contains the full definition of the geo feature record and vector spatial record. This is not mandatory, since not all fields are used.

The following data is encoded in the example (only ASCII data is shown):

Feature record identifier field	
RCID	0000000001
GRUP	002
OBJL	00018 (BOYSAW)
RVER	001

table A.8

Feature object identifier field	
AGEN	NL
FIDN	0000000001
FIDS	00001

table A.9

Feature record attribute field	
00004 (BOYSHP)	4
00075 (COLOUR)	3,1
00076 (COLPAT)	3
00116 (OBJNAM)	North sea 1

table A.10

Feature record national attribute field	
00301 (NOBJNM)	Noordzee 1

table A.11

The example is given for both the ASCII and binary implementation.

The size of the field length field (LR RP 20) and the size of the field position field (LR RP 21) are both set to 5 for all logical records.

The data description for the "0001" field is mandatory for use in an S-57 conforming file.

Both the ATTF and NATF fields are used. The "National attribute" [NATF] field uses lexical level 2 and contains the Dutch name for the buoy (Noordzee 1). Although lexical level 2 is not needed to encode the Dutch language, it is used in the example to clarify the use of a two-byte character set (see Part 3, clause 2.4 and Annex B).

Normally all information in an ISO/IEC 8211 conforming file is written without any demarcation (i.e. a stream of information with no line-feeds or CR's). In order to improve the readability of the examples, the elements of the interchange files are shown separately (with wrapped lines if applicable).

Substitutes for non-printable characters are defined in table A.12. These substitutes are only used in the examples. In an ISO/IEC 8211 file the true characters must be used. Explanatory text is shown in italics.

SPACE	(2/0)	□
UT (unit terminator)	(1/15)	Δ
FT (field terminator)	(1/14)	∇
NULL	(0/0)	•
binary data		▲

table A.12

A.4.1 ASCII example

DDR leader

```
019003LE1□0900319□!□5504
```

DDR directory

```
00000016300000000010004400163FRID0011400207FOID0007400321ATTF0006000395
NATF0006900450FFPC0008900524FFPT0008300613FSPC0008900696FSPT0009100785
VRID0008300876ATTV0005900959VRPC0007001018VRPT0007701088SGCC0005901165
SG2D0004601224SG3D0005101270ARCC0007801321AR2D0006001399EL2D0007401459
CT2D0004801533∇
```

DDR field area (field control field)

```
0000;&□□□Δ0001FRIDFRIDFOIDFRIDATTFFRIDNATFFRIDFFPCFRIDFFPTFRIDFSPCFRID
FSPT0001VRIDVRIDATTVVRIDVRPCVRIDVRPTVRIDSGCCVRIDSG2DVRIDSG3DVRIDARCCARC
CAR2DARCCEL2DARCCCT2D∇
```

DDR field area (data descriptive fields)

```
0100;&□□□ISO□8211□Record□IdentifierΔΔ (I(5))∇

1600;&□□□Feature□record□identifier□fieldΔRCNM!RCID!PRIM!GRUP!OBJL!RVER!
RUINΔ(A(2),I(10),A(1),I(3),I(5),I(3),A(1))∇

1600;&□□□Feature□object□identifier□fieldΔAGEN!FIDN!FIDSΔ(A(2),I(10),
I(5))∇

2600;&-A□Feature□record□attribute□fieldΔ*ATTL!ATVLΔ(I(5),A)∇

2600;&%/AFeature□record□national□attribute□fieldΔ*ATTL!ATVLΔ(I(5),A)∇

1600;&□□□Feature□record□to□feature□object□pointer□control□fieldΔFFUI!
FFIX!NFPTΔ(A(1),2I)∇

2000;&□□□Feature□record□to□feature□object□pointer□fieldΔ*LNAM!RIND!
COMTΔ(A(17),2A)∇

1600;&•••Feature•record•to•spatial•record•pointer•control•fieldΔFSUI!
FSIX!NSPTΔ(A(1),2I)∇

2000;&□□□Feature□record□to□spatial□record□pointer□fieldΔ*NAME!ORNT!
```

USAG!MASKΔ(A(12),3A(1))∇

1600;&□□□Vector□record□identifier□fieldΔRCNM!RCID!RVER!RUINΔ(A(2),
I(10),I(3),A(1))∇

2600;&□□□Vector□record□attribute□fieldΔ*ATTL!ATVLΔ(I(5),A)∇

1600;&□□□Vector□record□pointer□control□fieldΔVPUI!VPIX!NVPTΔ(A(1),2I)∇

2000;&□□□Vector□record□pointer□fieldΔ*NAME!ORNT!USAG!TOPI!MASKΔ(A(12),
4A(1))∇

1600;&□□□Coordinate□control□fieldΔCCUI!CCIX!CCNCΔ(A(1),2I)∇

2200;&□□□2-D□Coordinate□fieldΔ*YCOO!XCOOΔ(2R)∇

2200;&□□□3-D□Coordinate□fieldΔ*YCOO!XCOO!VE3DΔ(3R)∇

1600;&□□□Arc/Curve□definition□fieldΔATYP!SURF!ORDR!RESO!FPMFΔ(2A(1),
I(1),R,I)∇

2600;&□□□Arc□coordinate□fieldΔSTPT!CTPT!ENPT*YCOO!XCOOΔ(2R)∇

2600;&□□□Ellipse□coordinate□fieldΔSTPT!CTPT!ENPT!CDPM!CDPR*YCOO!XCOOΔ
(2R)∇

2600;&□□□Curve□coordinate□field_*YCOO!XCOO_(2R)∇

DR 1 (geo feature record) leader

00245□D□□□□□00109□□□5504

DR 1 directory

00010000600000FRID0002600006FOID0001800032ATTF0004100050NATF0002900091
FSPT0001600120∇

DR 1 field area

00001∇

FE0000000001P00200018001I∇

NL000000000100001∇

000044_000753,1_000763_00116North□sea□1Δ∇

00301•N•o•o•r•d•z•e•e•□•1•Δ•∇

VI0000000001NNN∇

DR 2 (vector spatial record) leader

00110□D□□□□□00067□□□5504

DR 2 directory

00010000600000VRID0001700006SG2D0002000023∇

DR 2 field area

00002∇

VI0000000001001I∇

52.10475Δ4.3004833Δ∇

A.4.2 Binary example

DDR leader

018833LE1□0900319□!□5504

DDR directory

000000163000000000010004300163FRID0010000206FOID0007000306ATTF0005900376
NATF0006800435FFPC0009000503FFPT0008600593FSPC0009000679FSPT0009000769
VRID0007800859ATTV0005800937VRPC0007100995VRPT0007601066SGCC0006001142
SG2D0004801202SG3D0005301250ARCC0007301303AR2D0006201376EL2D0007601438
CT2D0005001514∇

DDR field area (field control field)

0000;&□□□Δ0001FRIDFRIDFOIDFRIDATTFFRIDNATFFRIDFFPCFRIDFFPTFRIDFSPCFRID
FSPT0001VRIDVRIDATTVVRIDVRPCVRIDVRPTVRIDSGCCVRIDSG2DVRIDSG3DVRIDARCCARC
CAR2DARCCEL2DARCCCT2D∇

DDR field area (data descriptive fields)

0500;&□□□ISO□8211□Record□IdentifierΔΔ(b12)∇

1600;&□□□Feature□record□identifier□fieldΔRCNM!RCID!PRIM!GRUP!OBJL!RVER!
RUINΔ(b11,b14,2b11,2b12,b11)∇

1600;&□□□Feature□object□identifier□fieldΔAGEN!FIDN!FIDSΔ(b12,b14,b12)∇

2600;&-A□Feature□record□attribute□fieldΔ*ATTL!ATVLΔ(b12,A)∇

2600;&%/AFeature□record□national□attribute□fieldΔ*ATTL!ATVLΔ(b12,A)∇

1600;&□□□Feature□record□to□feature□object□pointer□control□fieldΔFFUI!
FFIX!NFPTΔ(b11,2b12)∇

2000;&□□□Feature□record□to□feature□object□pointer□fieldΔ*LNAM!RIND!
COMTΔ(B(64),b11,A)∇

1600;&□□□Feature□record□to□spatial□record□pointer□control□field∆FSUI!FSIX!NSPT∆(b11,2b12)∇

2000;&□□□Feature□record□to□spatial□record□pointer□field∆*NAME!ORNT!USAG!MASK∆(B(40),3b11)∇

1600;&□□□Vector□record□identifier□field∆RCNM!RCID!RVER!RUIN∆(b11,b14,b12,b11)∇

2600;&□□□Vector□record□attribute□field∆*ATTL!ATVL∆(b12,A)∇

1600;&□□□Vector□record□pointer□control□field∆VPUI!VPIX!NVPT∆(b11,2b12)∇

2000;&□□□Vector□record□pointer□field∆*NAME!ORNT!USAG!TOPI!MASK∆(B(40),4b11)∇

1600;&□□□Coordinate□control□field∆CCUI!CCIX!CCNC∆(b11,2b12)∇

2200;&□□□2-D□Coordinate□field∆*YCOO!XCOO∆(2b24)∇

2200;&□□□3-D□Coordinate□field∆*YCOO!XCOO!VE3D∆(3b24)∇

1600;&□□□Arc/Curve□definition□field∆ATYP!SURF!ORDR!RESO!FPMF∆(3b11,2b14)∇

2600;&□□□Arc□coordinate□field∆STPT!CTPT!ENPT*YCOO!XCOO∆(2b24)∇

2600;&□□□Ellipse□coordinate□field∆STPT!CTPT!ENPT!CDPM!CDPR*YCOO!XCOO∆(2b24)∇

2600;&□□□Curve□coordinate□field∆*YCOO!XCOO∆(2b24)∇

DR 1 (geo feature record) leader

00197□D□□□□□00109□□□5504

DR 1 directory

00010000300000FRID0001200003FOID0000900015ATTF0002900024NATF0002600053FSPT0000900079∇

DR 1 field area

▲▲∇

▲▲▲▲▲▲▲▲▲▲▲▲∇

▲▲▲▲▲▲••∇

▲▲4∆▲▲3,1∆▲▲3∆▲▲North□sea□1∆∇

▲▲•N•o•o•r•d•z•e•e•□•1•∆•∇

▲▲▲▲▲▲▲▲▲∇

DR 2 (vector spatial record) leader

00088□D□□□□□00067□□□5504

DR 2 directory

00010000300000VRID0000900003SG2D0000900012∇

DR 2 field area

▲▲∇

▲▲▲▲▲▲▲▲▲∇

▲▲▲▲▲▲▲▲▲∇

Page intentionally left blank

S-57 Part 3

Annex B - Alternate character sets

Edition 3.1

Page intentionally left blank

Important notice

All “Clarifications” in the latest edition of the Maintenance Document must be taken into account before making use of this document.

B. Alternate character sets

B.1 Implementation of alternate character sets in S-57

Text strings which are defined as “General Text” may be encoded using an alternate character set (see also clause 2.4). A number of different standards exist for the handling of general text. S-57 makes use of 3 lexical levels which support direct (multi-byte) coding. The three levels are described in table B.1.

level 0	ASCII text, IRV of ISO/IEC 646
level 1	ISO 8859 part 1, Latin alphabeth 1 repertoire (i.e. Western European Latin alphabet based languages.
level 2	Universal Character Set repertoire UCS-2 implementation level 1 (no combining characters), Base Multilingual plane of ISO/IEC 10646 (i.e. including Latin alphabet, Greek, Cyrillic, Arabic, Chinese, Japanese etc.)

table B.1

The ISO/IEC 8211 standard makes use of a data descriptive technique which facilitates the exchange of files containing data records between systems in a media-independent manner. It defines a generalized structure for a wide variety of data types and structures in terms of a Data Descriptive Record (DDR) which specifies the size and position of each data element within a data file. It also provides the means within the DDR for the description of the use of the contents of the data fields. The level of text data used in the data files can be described in the DDR. The default is ASCII (level 0). ISO 8211 provides two ways for the definition of a different lexical level; a lexical level can be set as the default for a complete file or as the default for a specific field type.

Since different fields within a S-57 data set may be encoded at different lexical levels, the mechanism within ISO/IEC 8211 to set a default for a specific field type is used for this Standard. Once a default is set for a specific field type within an ISO/IEC 8211 file, all instances of that field must be encoded at the same lexical level.

B.2 Implementation of alternate character sets in ISO/IEC 8211

When a character set other than the default ASCII character set (Lexical Level 0) is used for field types in a data set, ISO/IEC 8211 requires the following:

- DDR leader RP 7 must contain “E”
- (2/0)(2/1)(2/0) must be placed in DDR leader RP 17-19
- the Field Control Length field, DDR leader RP 10-11 must be set to “09"
- Field Controls RP 6-8 must contain one of the following truncated escape sequences:

Lexical level 0 (ASCII - ISO/IEC 646 IRV)	(2/0)	(2/0)	(2/0)
Lexical level 1 (Latin 1 - ISO 8859)	(2/13)	(4/1)	(2/0)
Lexical level 2 (Multilingual - ISO/IEC 10646)	(2/5)	(2/15)	(4/1)

- The unit terminator (UT) used to separate the subfields and the field terminator (FT) must be encoded in the character set used for the field in which they occur. Table B.2 defines the terminators for each level.

Lexical level	UT	FT
level 0	(1/15)	(1/14)
level 1	(1/15)	(1/14)
level 2	(0/0) (1/15)	(0/0) (1/14)

table B.2

level 2

B.3 Code tables

B.3.1 General

All text is defined in terms of character set code tables. Particular character codes are identified by a code table arranged into rows and columns in which 94 (or 96) character codes are assigned. A number of different character code tables has been in use internationally and these code tables are registered with ISO under ISO Standard 2375. ISO/IEC 10646 provides a comprehensive multilingual character set, which eliminates the need to select individual alphabets from the ISO registry. ISO/IEC 10646 contains as its base page the ISO 8859 part 1 Latin alphabet 1, which itself contains as its base the International Reference Version (IRV) alphabet ISO/IEC 646. ISO/IEC 646 (IRV) is equivalent to ASCII (American Standard Code for Information Interchange ANSI X3.4).

The alphabetic part of the ISO/IEC 646 (IRV) and ISO 8859 code tables is termed the Graphic or “G” set. Another specialized code table, the Control or “C0” set, is also defined. In addition to the G and C sets, two special characters are defined. These are ‘space’ and ‘delete’. Some of the C0 control characters are reserved for specialized use, such as transmission control in an asynchronous communications system or application level delimiting such as is used by ISO/IEC 8211. The only format effecting C0 characters required by S-57 are: Carriage Return (CR), Line Feed (LF), Back Space (BS), Horizontal Tab (HT), Vertical Tab (VT) and Form Feed (FF). Since S-57 operates in an 8-bit coding environment with three defined character repertoires corresponding to ISO standards, there is no need for the code extension characters Escape (ESC), Shift In (SI), or Shift Out (SO). All other C0 characters have a null meaning. The use of C0 characters may be further restricted by a relevant product specification.

The ASCII (ISO/IEC 646 IRV) code table caters largely to the needs of the English language. It defines 94 characters within a single 7-bit code table (with bit 8 zero in an 8-bit implementation). For other Latin languages where accented letters are used extensively, and for other alphabets, the International Organization for Standardization (ISO) has defined other standards. There are several different standards defined by ISO dependent on the size of the repertoire of characters which must be addressed.

The ISO 8859 standard uses bit 8 of an 8-bit character field to switch between two code tables, the ASCII code table on the left and a supplementary code table on the right containing 94 additional characters. Each character has a single code.

The ISO/IEC 10646 standard defines a “Universal Character Set” for virtually all languages in the world. To do this it must use 16-bits or more to identify each character. S-57 makes use of the Base Multilingual plane of ISO/IEC 10646 which uses 16-bits per character, handling ASCII, virtually all Latin alphabet languages, Greek, Hebrew, Cyrillic, Arabic, Chinese (Han - including Japanese Kanji and Korean Hangul), Japanese Katakana, etc. Virtually every modern alphabet is specified. Excluded are such character sets as ancient Egyptian hieroglyphics which require 32 bits per character.

The specific characters available under a given character set standard are called the repertoire of that standard. The defined repertoire of available characters is what is of principle importance for compatibility. Three levels of repertoire are defined for S-57 ranging from ASCII text to the support of any alphabet registered nationally or internationally. This range is broken into two broad levels: Basic Text and General Text.

Basic text (Level 0) is simply ASCII data and is used throughout this Standard for various purposes. The repertoire is simply the 94 characters defined in the ASCII character set plus the SPACE character plus specific C0 control characters (Carriage Return (CR) and Line Feed (LF), etc.). Other C0 characters are not used in basic text.

General text is used for the attribute fields (ATTF and NATF). Two levels of general text repertoire are defined. These levels have been defined to be efficient in various encodings and at different levels of usage. For example Level 1 general text makes use of the Latin Alphabet 1 repertoire (commonly called 8-bit ASCII) which is directly compatible with virtually all computer systems. Level 1 general text addresses the needs of Western European languages. Level 2 general text addresses the needs of almost all world languages, but it is less efficient in coding.

S-57 may be extended to a fourth level of general text which addresses the needs of all languages using the full capabilities of ISO/IEC 10646 UCS-4 requiring 4 bytes per character. However this approach can be very complex and inefficient and is reserved for further study at this time.

B.3.2 Level 0 Text Repertoire

The basic alphabet used in S-57 is the International Reference Version alphabet of ISO Standard 646, which is equivalent to ASCII. Table B.3 presents the Level 0 text repertoire, the Latin Alphabet Primary Code Table (ASCII). Both the G0 graphic and C0 control code tables are shown as well as the ‘space’ and ‘delete’ characters. Only the C0 format effecting characters are illustrated. All other C0 control codes are not used. The code extension characters from the C0 set (ESC, SI, and SO) are not used. The 7-bit code table is shown. Bit 8 in an 8-bit field is set to 0.
The delete character is only used in the update mechanism (see clause 8.4.2.2.a and 8.4.3.2.a).

				column								
				b8	0	0	0	0	0	0	0	0
				b7	0	0	0	0	1	1	1	1
				b6	0	0	1	1	0	0	1	1
				b5	0	1	0	1	0	1	0	1
row					0	1	2	3	4	5	6	7
b4	b2	b3	b1									
0	0	0	0	0			Space	0	@	P	`	p
0	0	0	1	1			!	1	A	Q	a	q
0	0	1	0	2			"	2	B	R	b	r
0	0	1	1	3			#	3	C	S	c	s
0	1	0	0	4			$	4	D	T	d	t
0	1	0	1	5			%	5	E	U	e	u
0	1	1	0	6			&	6	F	V	f	v
0	1	1	1	7			'	7	G	W	g	w
1	0	0	0	8	BS		(	8	H	X	h	x
1	0	0	1	9	HT		)	9	I	Y	i	y
1	0	1	0	10	LF		*	:	J	Z	j	z
1	0	1	1	11	VT		+	;	K	[	k	{
1	1	0	0	12	FF		,	<	L	\	l	|
1	1	0	1	13	CR		-	=	M	]	m	}
1	1	1	0	14			.	>	N	^	n	~
1	1	1	1	15			/	?	O	_	o	Delete
					C0		G0					

Latin Alphabet Primary Code Table (ASCII)

table B.3

B.3.3 Level 1 Text Repertoire

Table B.4 presents the Level 1 text repertoire, which is the 8-bit code table from ISO 8859 part 1. The G0 graphic portion is equivalent to ASCII. The same C0 code table is also used with the same restriction for the Format Effector characters. All other C0 control codes are not used. ISO 8859 part 1 also contains the ‘space’ and ‘delete’ character. The right hand side of the 8-bit coding environment contains the ISO 8859 supplementary code table and a blank C1 table. The supplementary characters are direct characters; that is, individual character codes are assigned for each accented character in the repertoire. In Lexical Level 0 and 1 each character is coded using a single character. There are no constructed characters. This simplifies processing of such data.
The delete character is only used in the update mechanism (see clause 8.4.2.2.a and 8.4.3.2.a).

				column	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
				b8	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
				b7	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
				b6	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
				b5	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
b4	b2	b3	b1	row																
0	0	0	0	0			Space	0	@	P	`	p			NBSP	°	À	Ð	à	ð
0	0	0	1	1			!	1	A	Q	a	q			¡	±	Á	Ñ	á	ñ
0	0	1	0	2			"	2	B	R	b	r			¢	²	Â	Ò	â	ò
0	0	1	1	3			#	3	C	S	c	s			£	³	Ã	Ó	ã	ó
0	1	0	0	4			$	4	D	T	d	t			¤	´	Ä	Ô	ä	ô
0	1	0	1	5			%	5	E	U	e	u			¥	µ	Å	Õ	å	õ
0	1	1	0	6			&	6	F	V	f	v			¦	¶	Æ	Ö	æ	ö
0	1	1	1	7			'	7	G	W	g	w			§	·	Ç	×	ç	÷
1	0	0	0	8	BS		(	8	H	X	h	x			¨	¸	È	Ø	è	ø
1	0	0	1	9	HT		)	9	I	Y	i	y			©	¹	É	Ù	é	ù
1	0	1	0	10	LF		*	:	J	Z	j	z			ª	º	Ê	Ú	ê	ú
1	0	1	1	11	VT		+	;	K	[	k	{			«	»	Ë	Û	ë	û
1	1	0	0	12	FF		,	<	L	\	l	|			¬	¼	Ì	Ü	ì	ü
1	1	0	1	13	CR		-	=	M	]	m	}			-	½	Í	Ý	í	ý
1	1	1	0	14			.	>	N	^	n	~			®	¾	Î	Þ	î	þ
1	1	1	1	15			/	?	O	_	o	Delete			¯	¿	Ï	ß	ï	ÿ
					C0		G0						C1		G1					

Latin Alphabet 1 Code Table ISO 8859-1

table B.4

B.3.4 Level 2 Text Repertoire

The Level 2 text repertoire supports all characters in the Base Multilingual Plane of ISO/IEC 10646, which is known as Universal Character Set-2 (for 2-byte coding). This character set is also known as "Unicode". It covers almost all languages in the world including the large iconographic character sets such as Chinese, Japanese and Korean. It is based on a 2-byte (16-bit) coding scheme.

There are several sub-repertoires defined in ISO/IEC 10646 UCS-2. For S-57, UCS-2 level 1 is used. UCS-2 level 1 is rigidly defined as one character per code (no combining characters).

The two byte coding scheme of UCS-2 addresses into a very large code table (or plane). UCS-4 uses four 8-bit codes (32 bits) to index into an enormous code space. The base plane of the UCS-4 code space, called the base multi-lingual plane, is equivalent to the UCS-2 code plane. The other planes support less frequently needed codes such as extended Chinese, Ancient Egyptian, etc. The first eight bits of the address space of both UCS-2 and UCS-4 match the ISO 8859 character set, and the first seven bits match ASCII. To the user of a database UCS codes are just 16 or 32 bit character codes. Direct support for ISO/IEC 10646 is provided in this manner by many computer vendors.

The entire repertoire of ISO/IEC 10646 is much too large to reproduce in this standard. Reference should be made to ISO/IEC 10646. In brief the alphabets supported are:

Collection number	Name	Code Position
1	Basic Latin + Space and Delete	0020 - 007F
2	Latin-1 Supplement	00A0 - 00FF
3	Latin Extended-A	0100 - 017F
4	Latin Extended-B	0180 - 024F
5	IPA Extensions	0250 - 02AF
6	Spacing Modifier Letters	02B0 - 02FF
7	Combining Diacritical Marks	0300 - 036F
8	Basic Greek	0370 - 03CF
9	Greek Symbols and Coptic	03D0 - 03FF
10	Cyrillic	0400 - 04FF
11	Armenian	0530 - 058F
12	Basic Hebrew	05D0 - 0EA
13	Hebrew Extended	0590 - 05CF, 05EB - 05FF
14	Basic Arabic	0600 - 0652
15	Arabic Extended	0653 - 06FF
16	Devanagari	0900 - 097F, 200C, 200D
17	Bengali	0980 - 09FF, 200C, 200D
18	Gurmukhi	0A00 - 0A7F, 200C, 200D
19	Gujarati	0A80 - 0AFF, 200C, 200D
20	Oriya	0B00 - 0B7F, 200C, 200D
21	Tamil	0B80 - 0BFF, 200C, 200D
22	Telugu	0C00 - 0C7F, 200C, 200D
23	Kannada	0C80 - 0CFF, 200C, 200D
24	Malayalam	0D00 - 0D7F, 200C, 200D
25	Thai	0E00 - 0E7F
26	Lao	0E80 - 0EFF
27	Basic Georgian	10D0 - 10FF
28	Georgian Extended	10A0 - 10CF
29	Hangul Jamo	1100 - 11FF
30	Latin Extended Additional	1E00 - 1EFF
31	Greek Extended	1F00 - 1FFF
32	General Punctuation	2000 - 206F
33	Superscripts and Subscripts	2070 - 209F
34	Currency Symbols	20A0 - 20CF
35	Combining Diacritical Marks for Symbols	20D0 - 20FF
36	Letter Like Symbols	2100 - 214F
37	Number Forms	2150 - 218F
38	Arrows	2190 - 21FF
39	Mathematical Operators	2200 - 22FF
40	Miscellaneous Technical	2300 - 23FF
41	Control Pictures	2400 - 243F
42	Optical Character recognition	2440 - 245F
43	Enclosed Alphanumerics	2460 - 24FF
44	Box Drawing	2500 - 257F

Collection number	Name	Code Position
45	Block Elements	2580 - 259F
46	Geometric Shapes	25A0 - 25FF
47	Miscellaneous Symbols	2600 - 26FF
48	Dingbats	2700 - 27BF
49	CJK Symbols and Punctuation	3000 - 303F
50	Hiragana	3040 - 309F
51	Katakana	30A0 - 30FF
52	Bopomofo	3100 - 312F
53	Hangul Compatibility Jamo	3130 - 318F
54	CJK Miscellaneous	3190 - 319F
55	Enclosed CJK Letters and Months	3200 - 32FF
56	CJK Compatibility	3300 - 33FF
57	Hangul	3400 - 3D2D
58	Hangul Supplementary-A	3D2E - 44B7
59	Hangul Supplementary-B	44B8 - 4DFF
60	CJK Unified Ideograms	4E00 - 9FFF
61	Private Use Area	E000 - F8FF
62	CJK Compatibility Ideograms	F900 - FAFF
63	Alphabetic Presentation Forms	FB00 - FB4F
64	Arabic Presentation Forms-A	FB50 - FDFF
65	Combining Half Marks	FE20 - FE2F
66	CJK Compatability Forms	FE30 - FE4F
67	Small Form Variants	FE50 - FE6F
68	Arabic Presentation Forms-B	FE70 - FEFE
69	Halfwidth and Fullwidth Forms	FF00 - FFEF
70	Specials	FFF0 - FFFD

table B.5

Page intentionally left blank