SEG RODE FORMAT
Record Oriented Data Encapsulation

SEG Technical Standards Committee
Subcommittee for High Density Media Formats

29 April 1997

RECORD ORIENTED DATA ENCAPSULATION FORMAT STANDARD

Prepared by the SEG Technical Standards Sub-Committee for High Density Media Formats

Eric Booth (Chairman)[1], Ugur Algan[2], Peter Duke[3], William Guyton[4], Mike Norris[5], Shane Stainsby[6], Jim Theriot[7], Barry Wildgoose[8], Don Wilhelmsen[9]

Contents

[1] RODE Consultants Ltd, Burnhouse, Milton Road, Lennoxtown, Strathclyde, G65 7NH, Scotland
Formerly BP Exploration, 4/5 Longwalk, Stockley Park, Uxbridge. Middlesex UB11 1BP England

[2] Petroleum Exploration Computer Consultants, Medway House, Lower Road, Forest Row, Sussex RH18 5HE, England;

[3] Simon Petroleum Technology, Horizon House, Azalea Drive, Swanley, Kent BR8 8JR, United Kingdom

[4] Western Geophysical , 3600 Briarpark Drive, Houston Texas 77042-4299, P.O.Box 2469 Houston Texas 77252-2469

[5] Western Geophysical , 3600 Briarpark Drive, Houston Texas 77042-4299, P.O.Box 2469 Houston Texas 77252-2469

[6] Schlumberger (Geco-Prakla), Schlumberger House, Buckingham Gate, Gatwick Airport, Gatwick, West Sussex RH6 0NZ, United Kingdom

[7] POSC, 1077 Westheimer, Suite 275, Houston, Texas 77042

[8] Mobil Exploration and Production Technical Center, P.O. Box 650232, Dallas Texas 75265-0232

[9] Schlumberger Well Services, Austin Systems Center, 8311 North FM Road, Austin, Texas 78726. P.O. Box 200015, 78720-0015

List of Figures

List of Tables

1. RODE Schema

Foreword

This schema was prepared by the Society of Exploration Geophysicists Technical Standards Sub-committee for high density media formats. This standard is administered by the Society of Exploration Geophysicists.

Background and Purpose

Digital data forms the basis of exploration geophysics. The formats used for acquiring and processing data were designed for very specific needs largely determined by the storage systems available. As a result these formats are discipline, system and application dependent. We archive and store this data for many years. The industry has several million seismic field tapes recorded on old media. These archives are growing rapidly with the increasing use of 3-D acquisition for exploration and appraisal.

The introduction of new high capacity media could reduce acquisition, processing, and storage costs and support automated access to the very large volumes of digital data. The processing industry needs to be able to move to the new media in stages protecting their investment in existing software and their ability to process data acquired on both new and old media. A standard encapsulation format provides a standard way of migrating old data into the new environment and allows new data to be recorded efficiently on new media using existing formats.

RP66 V2

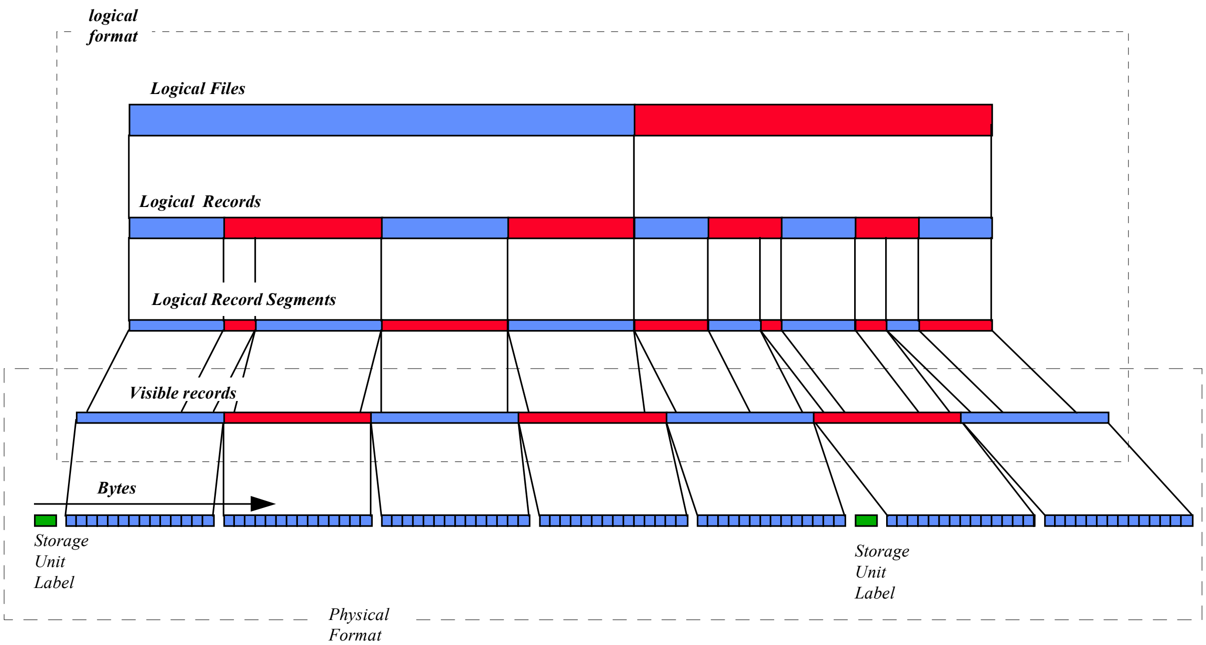
Recommended Practice 66, Version1 (RP66 V1) was published by the API in 1991 as a standard exchange format for digital well data. It incorporated the useful features of earlier formats and included the ability to define new objects and attributes. The primary features of RP66 V1 were machine independence, self description, semantic extensibility, and efficient handling of bulk data. A number of efforts by other oil industry groups identified RP66 as a suitable basis for a standard data exchange format. The major limitations of version 1 are its strong bias to well data and traditional media. RP66 version 2 removes the well log bias, provides better physical record structures and includes improvements identified by the users of version 1.

RP-66 has three distinct layers, schema describe user objects, attributes and standard values, the logical layer specifies the syntax used to manage these objects and the physical layer contains generic rules for building physical data records and media bindings for each specific medium. RP-66 assumes that sequential logical records exist within a logical file, the mapping of these data to a physical media (either as physical files or streams of data) is defined by RP-66.

The RP-66 logical format describes how to encode data in a media independent sequence of 8 bit bytes. At the highest level, the logical format is a sequence of logical files. A logical file is divided into a sequence of related logical records. A logical record contains a set of related values and is subdivided into a sequence of logical record segments. A logical record segment is contained in a visible record. A visible record has a header and trailer and may contain one or more logical record segments. A visible record consists of a sequence of bytes mapped into some physical medium.

Figure 1 shows the transition from the media dependent organisation of data (the physical format) to the media independent organisation of data (the logical format). A single physical binding will be defined for each media to provide an optimum compromise between storage efficiency and data security. RP66 logical records contain either sets of related objects in Explicitly Formatted Logical Records (EFLRs) or binary data in Indirectly Formatted Logical records (IFLRs) whose format is defined by one of the preceding objects. The combination of these two record types provides an efficient way of recording descriptive and parameter data with large volumes of digital data in a highly structured but not restrictive manner.

Figure 1 Conceptual view of logical format (bound to physical format)

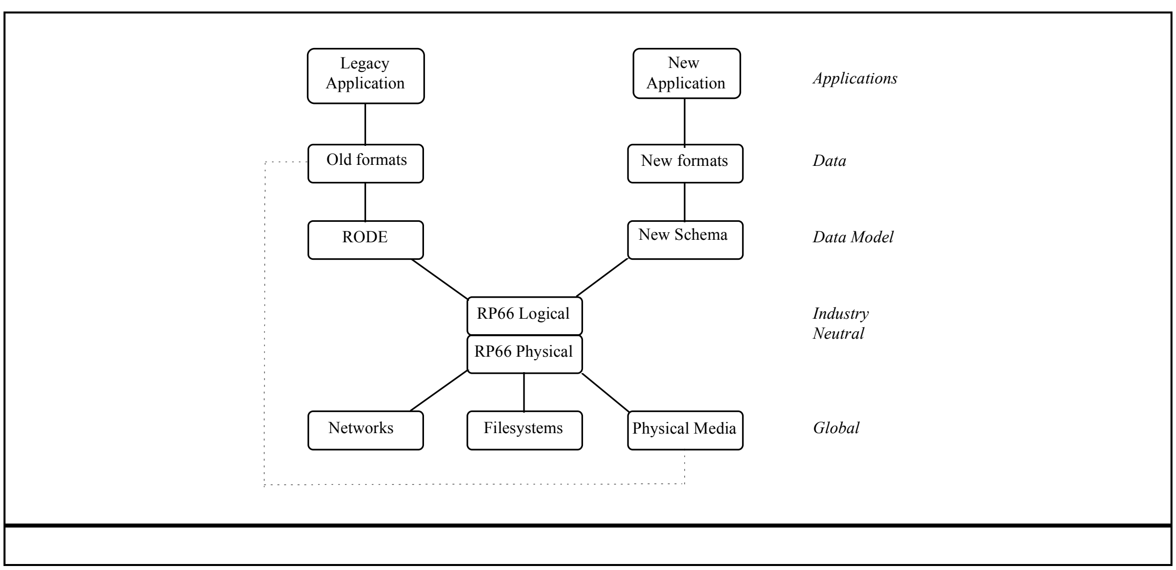


Layers of standards.

The oil industry is one of the largest users of data storage systems. These systems are developed by other industries and the standards for these systems are agreed on a global basis. The integration of data and systems within our industry needs common data storage formats and data management structures.

Figure 2 shows a schematic view of the relationship between standards. Global standards are specified by international institutions. The API RP66 Exploration & Production Data Digital Interchange Standard provides an industry neutral means of encoding data using global media, file and network standards. RP66

Figure 2 Layer structure of standards



data models are defined as schema that allow applications to access the data. RODE is one such schema that is designed for storing and retrieving data in existing formats using RP66.

Acknowledgements.

This schema is based on proposals developed by the ad-hoc formats and standards committee formed at the Seis-Media meeting held at BPX Stockley Park in April 1993.
The committee thank BP and Mobil for the use of their video-conference facilities.
The chairman thanks BP, Mobil and Statoil for their support and funding to complete the work on this schema.

Organisation of SEG RODE

This document is divided into two parts and an appendix. These are described below together with the edition of each that is current with the publication of this edition of the current part**.**

1. Part 1: The RODE schema, first edition 16 August 1995.
2. Part 2: The RODE schema dictionary, first edition 16 August 1995.
3. Appendix A: Example of encapsulation of two SEGY files.

1.1 Introduction

This standard contains introductory, normative and annotative information. Introductory information comprises everything that precedes section 1.2, Scope. The introductory clauses describe how the standard is organised, and provide background information on the development of this and related standards. Normative information defines the requirements that must be satisfied by any data conforming to this standard. Annotative information contains the rationale for and illustration of the normative information. The standard is completely stated by the normative information.

Different styles are used to distinguish between normative and annotative information.
All normative information is written using this font , and is contained either in numbered paragraphs or numbered tables. All figures are annotative, and all annotative text is written in unnumbered paragraphs in italic font and indented as shown here:

This is a paragraph of annotative text. Its purpose is to include informal comments on the normative information.

1.2 Scope

This section describes the SEG RODE schema using the methodology defined in part 1 of RP66 version 2. RP66 provides media independence. The RODE schema is designed to address three problem areas facing the geophysical industry:

a) *There are several new media technologies, we need a single high level format that can be used on any media.*
b) *There is a need to store more metadata (data that identifies who, what, why, when, where) with the data.*
c) *The volumes of data and the number of files managed are increasing exponentially. There is a need to have a standard method of storing media index and dataset cross reference information.*

RP66 provides media independence. The RODE schema is designed to encapsulate data in existing formats based on the variable length record and filemark facilities of standard 1/2" magnetic tape.

The schema retains all other status information, if necessary, generated from original tapes during transcription to new media.

The RODE-CONTEXT and RODE-ANCILLARY-INFORMATION objects handle minimal metadata. The schema also allows a direct link to other metadata. The RODE-DIRECTORY-xxxx objects provide basic indexing and cross-referencing functions.

1.3 Definitions

1. **attribute:** A named item of information or data pertaining to an object.
2. **attribute count:** The number of elements in an attribute value.
3. **attribute label:** The name of an attribute.
4. **attribute representation code:** a code that identifies the recorded representation of each element of an attribute value.
5. **attribute units:** An expression that represents the units of measurement of each element of an attribute value.
6. **channel:** One or more sequential data values
7. data descriptor reference: The name of an object written at the beginning of an IFLR and used to associate the IFLR with the object that describes its contents.
8. **data model:** A description of a specification and representation paradigm for data.
9. **dictionary:** A database in which identifiers and reference values used under RP66 are maintained and administered.
10. **encapsulation:** A process whereby data in one format is packaged with headers, trailers and metadata so that it can be written in another format.
11. **EFLR:** Explicitly Formatted Logical Record: The content of an EFLR, that consists of a set of objects, is determined from analysis of the record itself.
12. **filemark:** A special record or state of a media index that indicates the end of a physical file.
13. **IFLR:** Indirectly Formatted Logical record: The content of an IFLR is determined from analysis of elated EFLRs
14. **logical file:** The main unit of data exchange. It consists of a sequence of one or more logical records, beginning with a record containing a single file header object
15. **logical record:** An organisation of data values into coherent, semantically related packets of information. A logical record may have any length greater than sixteen bytes and is organised in one of two syntactic forms: explicitly-formatted-logical records or indirectly-formatted-logical-records.
16. **logical record segment:** A construct that contains the structure necessary to describe and support the physical implementation of a logical record. A logical record is implemented as one or more logical record segments. A segment is wholly contained in a visible record, but two segments of the same logical record may be in different visible records.
17. **media binding:** A description of how RP66 visible records are organised for a specific media.
18. **metadata:** Information about the data that allows users to identify and understand the data.
19. **object:** An instance of a specific object type.
20. **object type:** A logical entity of a schema that has a unique type name and one or more defined attributes. Instances of an object type are written in explicitly formatted logical records.
21. **organisation code:** A number assigned by the American Petroleum Institute to an organisation that identifies the organisation and represents schemas and dictionaries defined and administered by the organisation.
22. **partition:** An independent recording area resulting from physical formatting of the media that can be mounted as though it were a single volume.
23. **schema:** A formalised description of the encoding of information defined by a logical model, typically in terms of a data model.
24. **storage set:** An ordered set of RP66 storage units, the position in the set is specified in the RP66 storage unit label.

25. **storage unit:** A logical volume of RP66 data containing one or more RP66 logical files. All data on an RP66 storage unit must be recorded using the same revision of RP66.
26. **visible record:** The interface between the logical format and a medium-specific physical format. A visible record has a header, a body, and a trailing length.

1.4 Audience

This document is intended for:

a) Original developers and maintainers of software
b) Data administrators
c) It is assumed that the reader has a basic understanding of API RP66.

1.5 References

American Petroleum Institute, The October 1994 Committee draft of API Recommended Practice 66, V2, Second Edition, XXXX, 1994 "Recommended Practice for Exploration & Production Data Digital Interchange", catalog number G66002

To order the RP66 standard contact:

American Petroleum Institute
Publication Department
1220 L Street NW
Washington DC, District of Columbia 20005
Telephone: (202) 682-8375
Fax: (202) 682-8537

1.6 Authority

Changes to the SEG RODE schema are recommended by the SEG Technical Standards High Density Media sub-committee (RP66 Organisation Code 463) and approved by the SEG Technical Standards committee and then the SEG Executive committee.

The intent is to maintain as much consistency as possible between editions. The principle motivation for a new edition will be to add new object types or attributes.

1.7 Concepts

The RODE schema is a collection of object types specifically designed for the encapsulation of data created using existing geophysical standard formats. These datasets may range from a few hundred bytes to many terabytes. The very large datasets (normally pre-stack 3D seismic data) are usually divided into subsets (typically seismic lines) consisting of many ensembles of variable length records separated by filemarks.

The data model represented by the RODE schema is implied by the descriptions of the object types. The RODE schema provides the following object types for data management:

a) RODE-CONTEXT - used to identify how, when and where the file was created
b) RODE-DIRECTORY-SCOPE - used as a header reference point for a directory of files
c) RODE-DIRECTORY-ENTRY - a generic directory entry that can be stored on the high capacity media

And the following for data storage:

a) RODE-ANCILLARY-INFORMATION - used to store unformatted metadata about a file (e.g. data from a label attached to an input tape)
b) CHANNEL and FRAME - additional channels are defined for the RP66 basic schema so that these industry neutral concepts may be used for encapsulation
c) RODE-SUMMARY - provides a check on the contents of the file

1.8 Dictionary controlled identifiers

The following object types have dictionary controlled identifiers:

a) RODE-CONTEXT
b) RODE-DIRECTORY-SCOPE
c) RODE-DIRECTORY-ENTRY

1.9 Required versus optional use of attributes

Use of any attribute is considered optional unless otherwise stated.

1.10 Frequently used attributes

Attributes DESCRIPTION and EXTENDED-ATTRIBUTES are used in all object types as specified in the RP66 basic schema. The restrictions on the values of these attributes are shown in Table 1. The tables are in the same format as that used in RP66 where "c" specifies the number of values allowed, "r" specifies the representation code and "u" specifies the units ("u=" specifies that this attribute is unitless.).

Table 1 Frequently used attributes

Attribute Label	Restrictions	Note
DESCRIPTION	c=1, r=ascii, u=	1
EXTENDED-ATTRIBUTES	r=objref, u=	2

Notes:

1. DESCRIPTION is a human readable textual description of the object and is not intended to impose or posses syntactic or semantic content. It is available in all RODE schema object types and has the same meaning everywhere.

2. EXTENDED-ATTRIBUTES is a list of references to other objects, typically in other schemas, that contain additional privately-defined attributes that apply to a given object. It is available for all RODE schema objects and has the same meaning everywhere.

1.11 Required Object types

1.11.1 RODE-CONTEXT

The RODE-CONTEXT object provides a framework for all the metadata specifically associated with encapsulation and data management.

Table 2 shows the RODE-CONTEXT attributes and their properties.

Table 2 RODE-CONTEXT Attributes

Attribute Label	Restrictions	Mandatory	Note
INTERNAL-FORMAT	c=1, r=ident, u=	Yes	1
OPERATING-SYSTEM	c=1, r=ascii, u=	Yes	2
PROGRAM	r=ascii, u=	Yes	3
PROGRAM-VERSION	r=ascii, u=	Yes	4
AUTHOR-CODE	c=1, r=ulong, u=	No	5
AUTHOR-NAME	c=1, r=ascii, u=	Yes	6
PRODUCER-CODE	c=1, r=ulong, u=	No	7
PRODUCER-NAME	c=1, r=ascii, u=	Yes	8
PRODUCER-SYSTEM	c=1, r=ascii, u=	Yes	9
PRODUCER-DRIVE	c=1, r=ascii, u=	Yes	10
CLIENT-COMPANY	r=ascii, u=	Yes	11
META-DATA-ROOT	c=1, r=objref, u=	No	12
SOURCE-COMPANY	r=ascii, u=	No	13
SOURCE-PRODUCER-CODE	c=1, r=ulong, u=	No	14
SOURCE-PRODUCER-NAME	c=1, r=ascii, u=	No	15
SOURCE-COMMENTS	c=1, r=ascii, u=	No	16
SOURCE-CREATION-TIME	c=1, r = dtime	No	17
SOURCE-IDENTIFIER	c=1, r=ident, u=	No	18
SOURCE-MEDIA	c=1, r=ident, u=	No	19

Notes:
A RODE-CONTEXT object is referenced by the context attribute of one or more origin objects and provides contextual information about data in the logical file having the same origins.

Items 13-19 are provided for encapsulation of data copied from existing media.

1. The INTERNAL-FORMAT attribute contains the name of the Format of the data that has been encapsulated. This is a required attribute, it is not dictionary controlled but the recommended names for common formats in the RODE schema dictionary must be used where applicable. If the input format is unknown at the time of encapsulation, this must be explicitly stated as UNDETERMINED.
2. The OPERATING-SYSTEM attribute should clearly identify the operating system used to create the file including version numbers where appropriate.
3. The PROGRAM attribute is the name of the program used to generate the RODE data. There is no restriction on the count for this attribute as it is possible to have several programs linked together to produce a RODE file.
4. The PROGRAM-VERSION is the author's description of the version of their program used to create the logical file.
5. The AUTHOR-CODE attribute is the API code of the organisation that wrote the encapsulation software. This field may be absent if the organisation does not have a code.
6. The AUTHOR-NAME attribute is the organisation name that wrote the RODE implementation.
7. The PRODUCER-CODE attribute is the API code for the organisation that created this logical file. This field may be absent if the organisation does not have a code.
8. The PRODUCER-NAME attribute is the name of the organisation that created this logical file.
9. The PRODUCER-SYSTEM attribute is the host identifier for the system that created the data.
10. The PRODUCER-DRIVE attribute is the host computer's identification for the drive that created this file
11. The CLIENT-COMPANY attribute is the name of the client company or companies for whom the output data was written.

12. The META-DATA-ROOT attribute is the object reference of another object containing RP66 metadata related to this RODE dataset.
13. The SOURCE-COMPANY attribute is the name of the client company for whom the data was originally acquired.
14. The SOURCE-PRODUCER-CODE attribute is the API code (where available) of the company that created the original data set.
15. The SOURCE-PRODUCER-NAME attribute is the name of the company that created the original dataset.
16. The SOURCE-COMMENTS attribute allows the inclusion of any additional metadata for subsequent cataloguing and interpretation of stored data.
17. The SOURCE-CREATION-TIME attribute is the time and date that the original dataset was created.
18. The SOURCE-IDENTIFIER attribute is the user's identification for the original media (the tape number).
19. The SOURCE-MEDIA attribute is the generally accepted name for the media that was used to store the original data. e.g. 800 bpi 1/2" tape will be coded as 800-BPI . The RODE schema dictionary contains a list of recommended names of common media. This attribute is not dictionary controlled to allow the specification of old or special media types. If the media is in the RODE standard list (see the RODE schema dictionary) the specified name must be used.

1.12 Optional object types

1.12.1 RODE-ANCILLARY-INFORMATION

The Ancillary-Information Object provides a means of storing metadata without reference to an external model. It allows the recording of all information about the dataset that is available at the time of encapsulation in a structured manner.

Table 3 shows the attributes and their properties of the RODE-ANCILLARY-INFORMATION object.

Table 3 RODE-ANCILLARY-INFORMATION attributes

Attribute Label	Restrictions	Mandatory	Note
DESIGNATION	c=1, r= ascii	Yes	1
VALUE		Yes	2

Notes:

1. The ANCILLARY-INFORMATION object is not mandatory, but if used then both attributes are required for each object.
2. The DESIGNATION attribute must provide sufficient identification of the information for future reference.
3. The VALUE attribute contains the actual value or values assigned to this attribute.

1.12.2 RODE-SUMMARY

The RODE-SUMMARY object provides a means of storing summary information about the data that has been encapsulated. This is useful to double check the integrity of the copy process. Table 4 shows the attributes and properties of the RODE-SUMMARY object.

Table 4 RODE-SUMMARY attributes

Attribute Label	Restrictions	Mandatory	Note
NUMBER-OF-BYTES-HIGH	c=1, r=ulong, u=	No	1
NUMBER-OF-BYTES-LOW	c=1, r=ulong, u=	No	1
NUMBER-OF-RECORDS	c=1, r=ulong, u=	Yes	2
NUMBER-OF-FILES	c=1, r=ulong, u=	Yes	3
NUMBER-OF-ERRORS	c=1, r=ulong, u=	No	4
NUMBER-OF-EOTS	c=1, r=ulong, u=	No	5
NUMBER-OF-EOFS	c=1, r=ulong, u=	No	6

Notes:

1. The NUMBER-OF-BYTES-HIGH attribute and the NUMBER-OF-BYTES-LOW attribute allow the total number of bytes of the encapsulated data to be saved for future reference. The total number of bytes read is : NUMBER-OF-BYTES-HIGH * 1024 * 1024 *1024 + NUMBER-OF-BYTES-LOW.
2. The NUMBER-OF-RECORDS attribute is the number of data records read from the encapsulated data and transferred to the output.
3. The NUMBER-OF-FILES attribute is the number of input data files transferred to the output, this does not include any zero length files resulting from two or more consecutive end of file marks on the input media.
4. The NUMBER-OF-ERRORS attribute is a count of the number of unrecovered errors on the input data stream where a RODE-ERROR condition was recorded.
5. The NUMBER-OF-EOTS attribute is the number of End Of Media warnings that have been included in this file, note that a double EOF is not considered to be equivalent to an EOT for RODE.
6. The NUMBER-OF-EOFS attribute is the total number of EOF markers recorded in the encapsulated file.

1.12.3 RODE-DIRECTORY-SCOPE

The RODE-DIRECTORY-SCOPE object acts as a header reference point for a RODE directory. Table 5 shows the properties of the single attribute for the RODE-DIRECTORY-SCOPE object.

Table 5 RODE-DIRECTORY-SCOPE attributes

Attribute Label	Restrictions	Mandatory	Note
DIRECTORY-SCOPE	c=1, r=ident, u=	Yes	1

Notes:

1. The DIRECTORY-SCOPE attribute specifies the reason the directory was created, for example
   FILE-SET-DIRECTORY for all files in a File Set.
   STORAGE-UNIT-DIRECTORY for all files on a Storage Unit.
   STORAGE-SET-DIRECTORY for all files on a Storage Set.
   PROCESSING-HISTORY-DIRECTORY for all files used to generate this file-set

   If a directory is created for another reason it should be stated in a similar form. RODE directories provide a structure whereby a disjoint set of files may be linked together and managed as a unit. Please contact the SEG with suggestions for additional directory types that can be issued as a minor revision to this document.

We recommend that a RODE-STORAGE-UNIT directory should be written on any high capacity storage unit that is archived, retention of this information with the media will greatly simplify future data management.

1.12.4 RODE-DIRECTORY-ENTRY

The RODE-DIRECTORY-ENTRY object is a RODE specific object provided to allow the storage of media management information within the RP66 format. The attributes of the RODE-DIRECTORY-ENTRY object are given inTable 6:

Table 6 RODE-DIRECTORY-ENTRY attributes

Attribute Label	Restrictions	Mandatory	Note
RODE-DIRECTORY-SCOPE	r = obname, u=	Yes	1
FILE-SET-NAME	c=1, r=ascii, u=	Yes	2
FILE-SET-NUMBER	c=1, r=ident, u=	Yes	3
PRODUCER-CODE	c=1, r=ulong, u=	No	3
PRODUCER-NAME	c=1, r=ident, u=	Yes	4
STORAGE-SET-ID	c=1, r=ident, u=	Yes	5
VOLUME-SERIAL-NUMBER	c=1, r=ident, u=	Yes	6
FILE-SEQUENCE-NUMBER	c=1, r=ascii, u=	Yes	7
FILE-ID	c=1, r=ascii, u=	Yes	8
FILE-NUMBER	c=1, r=ulong, u=	Yes	9
CREATION-DATE	c=1, r = dtime, u=	Yes	10
MEDIA-TYPE	c=1, r=ident, u=	Yes	11
PARTITION	c=1, u=	media dependent	12
PHYSICAL-FILE-NUMBER	c=1, r=ulong, u=	media dependent	13
PHYSICAL-ADDRESS		media dependent	14

Notes:

1. The RODE-DIRECTORY-SCOPE attribute contains the object names of the RODE-DIRECTORY-SCOPE objects that define the directories. A RODE directory is the collection of all RODE-DIRECTORY-ENTRY objects that have a common RODE-DIRECTORY-SCOPE attribute. A single RODE-DIRECTORY-ENTRY may be a member of more than one RODE directory.

   The remaining attributes of this object refer to the file being described. Their values are either copies of the values of attributes of the file's defining origin or its RODE-CONTEXT (if appropriate) or they are values defining the location of the file.

2. The FILE-SET-NAME attribute matches the file-set-name of the file's defining origin object, i.e. the first origin object in the file.
3. The FILE-SET-NUMBER attribute is the file set number from the file's defining origin.
4. The PRODUCER-CODE and PRODUCER-NAME attribute are copied from the rode-context object.
5. The STORAGE-SET-ID attribute is the storage set identifier sub-field from the storage unit label.
6. The VOLUME-SERIAL-NUMBER attribute is the volume serial number sub-field of the storage unit label.
7. The FILE-ID attribute is copied from the file-header object and contains a descriptive identifier for the file.
8. The FILE-SEQUENCE-NUMBER attribute is the positive integer taken from the file’s file-header object that indicates the relative sequential position of the file in the storage set. This number is recorded in each visible record and provides a means of navigating through a storage set using RP66 data.
9. The FILE-NUMBER attribute is the sequential logical file number within the file-set.
10. The CREATION-DATE is the date and time when a logical file was created.

11. The MEDIA-TYPE attribute is the accepted name of the media including density. The RODE schema dictionary contains a list of current media and density options with recommended names. The names in the list should be used for the media designated, other media should be given their accepted commercial name and the committe advised that the schema is being used for this media.
12. The PARTITION must be specified if the media supports partitions.
13. The PHYSICAL-FILE-NUMBER attribute is the position of the physical file on the media as seen by the device sub-system for the physical file. For some media this will be recorded on a separate track, for other media it is the sequential file number and is the only method for faster access to a file (i.e. forward space over N previous files).
14. The PHYSICAL-ADDRESS attribute allows rapid access to the physical file using the appropriate scheme for the specific media. The value of this attribute will be media specific and is only meaningful in conjunction with the media type. Many helical scan media provide absolute address tracks that can be read whilst the media is moving at high speed. Direct addressing can be to a specific set of tracks on a Helical Scan system or to a track and block on a Serpentine system. The interpretation of this field will depend on the media specified by the MEDIA-TYPE attribute. For some media this is not an absolute address so the user (or the device driver) should always check that the media is correctly positioned at the start of the file.

RECORD ORIENTED DATA ENCAPSULATION FORMAT STANDARD

2. RODE Schema: Dictionary

Foreword

This schema was prepared by the Society of Exploration Geophysicists Technical Standards Sub-committee for high density media formats. This standard is administered by the Society of Exploration Geophysicists.

Background and Acknowledgements.

This schema is based on proposals developed by the ad-hoc formats and standards committee formed at the Seis-Media meeting held at BPX Stockley Park in April 1993.
The committee thank BP and Mobil for the use of their video-conference facilities.
The chairman thanks BP, Mobil and Statoil for their support and funding to complete the work on this schema.

Organisation of SEG RODE

This document is divided into two parts and an appendix. These are described below together with the edition of each that is current with the publication of this edition of the current part**.**

1. Part 1: The RODE schema, first edition 16 August 1995.
2. Part 2: The RODE schema dictionary, first edition 16 August 1995.
3. Appendix A: Example of encapsulation of two SEGY files.

2.1 Introduction

This standard contains introductory, normative and annotative information. Introductory information comprises everything that precedes section 2.2, Scope. The introductory clauses describe how the standard is organised, and provide background information on the development of this and related standards. Normative information defines the requirements that must be satisfied by any data conforming to this standard. Annotative information contains the rationale for and illustration of the normative information. The standard is completely stated by the normative information.

Different styles are used to distinguish between normative and annotative information. All normative information is written using this font. All figures are annotative, and all annotative text is written in unnumbered paragraphs in italic font and indented as shown here:

> *This is a paragraph of annotative text. Its purpose is to include informal comments on the normative information.*

2.2 Scope

This part lists and describes reference values for attributes belong to the SEG Record Oriented Data Encapsulation (RODE) schema (see part 1).

2.3 References

American Petroleum Institute, The October 1994 Committee draft of API Recommended Practice 66, V2, Second Edition, XXXX, 1994 "Recommended Practice for Exploration & Production Data Digital Interchange", catalog number G66002

To order the RP66 standard contact:

American Petroleum Institute
Publication Department
1220 L Street NW
Washington DC, District of Columbia 20005
Telephone: (202) 682-8375
Fax: (202) 682-8537

2.4 Authority

Changes to the SEG RODE schema dictionary are recommended by the SEG Technical Standards High Density Media (RP66 Organization Code 463) and then the SEG Executive committee. A new edition of the RODE schema dictionary occurs when approved by the SEG Executive Committee and has an edition number one greater than the previous edition number.

2.5 Concepts

The tables presented here contain reference values defined under the RODE schema for attributes of either the API basic schema or the RODE schema. There is one table per attribute. Each attribute is identified by its label preceded by the object type to which it belongs separated by a colon (:). Objects belonging to the basic schema are identified by the prefix ‘0/’ denoting its schema code. Tables are presented alphabetically by object type first and attribute second.

All RODE schema reference values use representation code IDENT if written in RODE schema objects and TIDENT if written in API basic schema objects.

2.6 0/CHANNEL reference values

The channels are actually recorded in a sequence of IFLRs and the order of the channels within each IFLR is specified by a FRAME Object. RODE channel objects are not dictionary controlled, a suite of specific channels are defined in this dictionary but users may define additional channels as required. Any such channels must have a 0/CHANNEL:KIND attribute selected from Table 7.

2.6.1 KIND

The kinds of channels and the associated restrictions are specified in Table 7 . CHANNEL object defines and characterises channels, RP66 allows many different types of channels to be specified.

Table 7 0/CHANNEL:KIND reference values and restrictions

0/CHANNEL:KIND Reference value	**Representation Code**	**Mandatory**	**Note**
RODE-DATA	USHORT	YES	1
RODE-ERROR	STATUS	NO	2
RODE-END-OF-FILE	STATUS	NO	3
RODE-STATUS		NO	4

Notes:

1. RODE-DATA is an explicitly-sized channel where the length of the data record (in bytes) is included in each IFLR frame. Variable length encapsulated data are stored as unsigned bytes. RP66 variable length channels support zero length instances of the channel. The RODE-DATA channel may contain data even though the RODE-ERROR channel is true.
2. RODE-ERROR consists of a single RP66 status byte where FALSE (all bits 0) signifies that no error was detected during the original read or data creation operation. If an unrecovered error was detected during the creation of the RODE-DATA then the status should be TRUE (bit 8 =1), any data available for the record should be written in the RODE-DATA channel, it will be the decoding application’s responsibility to decide what to do with these data.

3. RODE-END-OF-FILE is an RP66 status byte that indicates an End of File marker. In the RODE schema this is a specific separate event and hence an IFLR where RODE-END-OF-FILE is true must not contain any data, (i.e. the length of the RODE-DATA channel must be zero).
4. A RODE-STATUS channel can be used for any other status information associated with a specific RODE-DATA channel.

The RODE-DATA channel provides a means of specifying several variable length channels in a single frame. Three component seismic could be recorded into IN-LINE, CROSS-LINE and VERTICAL channels.

2.6.2 Channel object names.

The 0/CHANNEL:KIND values may be used as CHANNEL names. Any such channel will be assumed to have the same value for its KIND attribute. Table 8 contains a list of object names specified by this schema, these objects have specific meanings that relate to existing seismic data formats and acquisition practices.

The use of these object names implies the 0/CHANNEL:KIND attribute shown in the third column and the appropriate restrictions shown in Table 7.

Table 8 0/CHANNEL object names

0/CHANNEL object names Reference value	Representation Code	Implicit KIND	Note
RODE-DATA	USHORT	RODE-DATA	1
RODE-ERROR	STATUS	RODE-ERROR	2
RODE-END-OF-FILE	STATUS	RODE-END-OF-FILE	3
RODE-END-OF-TAPE	STATUS	RODE-STATUS	4
RODE-BEGINNING-OF-TAPE	STATUS	RODE-STATUS	5
RODE-BLANK-TAPE	STATUS	RODE-STATUS	6
RODE-PARITY-ERROR	STATUS	RODE-ERROR	7
RODE-CHECKSUM-ERROR	STATUS	RODE-ERROR	8
RODE-INPUT-DRIVE-STATUS	USHORT	RODE-STATUS	9
RODE-RECOVERED-ERROR	STATUS	RODE-STATUS	10
RODE-SEQUENTIAL-FILE-NUMBER	ULONG	RODE-STATUS	11
RODE-SEQUENTIAL-RECORD-NUMBER-IN-FILE	ULONG	RODE-STATUS	12
RODE-SEISMIC-HEADER	USHORT	RODE-DATA	13
RODE-SEISMIC-TRACE	USHORT	RODE-DATA	14

Notes:

1. RODE-DATA is an explicitly-sized channel where the length of the data record (in bytes) is included in each IFLR frame. Variable length encapsulated data are stored as unsigned bytes. RP66 variable length channels support zero length instances of the channel. The RODE schema requires a zero length RODE-DATA channel when any of RODE-EOF, RODE-BEGINNING-OF-TAPE or RODE-BLANK-TAPE are true. The RODE-DATA channel may contain data even though the RODE-ERROR channel is true.
2. RODE-ERROR consists of a single RP66 status byte where FALSE (all bits 0) signifies that no error was detected during the original read or data creation operation. If an unrecovered error was detected during the creation of the RODE-DATA then the status should be TRUE (bit 8 =1), any data available for the record should be written in the RODE-DATA channel, it will be the decoding application’s responsibility to decide what to do with these data.

3. RODE-END-OF-FILE is an RP66 status byte that indicates that an End of file marker was read from the input media, i.e. the application program should be notified that an EOF marker has been read. An IFLR containing a RODE-END-OF-FILE must not contain any data, (i.e. the length of the RODE-DATA channel must be zero).
4. RODE-END-OF-TAPE is an RP66 status byte that indicates that an End Of Media warning occurred on the input media. There normally will be data in the RODE-DATA channel. The RODE-END-OF-TAPE does not imply or require the RODE-END-OF-FILE status to be true. Many media allow data to be written after the EOT warning.
5. RODE-BEGINNING-OF-TAPE is an RP66 status byte that indicates that a beginning of media condition was reported for the input media. Many formats write a different record at the beginning of each volume and this status provides a double check that the next record is the first on an input media volume.
6. RODE-BLANK-TAPE is an RP66 status byte provided for the encapsulation of seismic field data where blank areas may exist if brand new tapes were used and physical gaps left on the tape. Blank tape can exist at the end of a seismic field dataset or where the field system was moved, in this case data may exist after the blank tape. Since no data has been read , the RODE-DATA channel length must be zero.
7. RODE-PARITY-ERROR is an RP66 status byte indicating that an unrecovered parity error was detected on the input by the creating system. Any data that is available for this record should be written in the RODE-DATA-CHANNEL. The end-user can then decide whether or not to use the data.
8. RODE-CHECKSUM-ERROR is an RP66 status byte indicating that a checksum error was detected in the input data stream.
9. RODE-INPUT-DRIVE-STATUS is a copy of the full input unit status for those systems capable of reporting detailed status conditions. The length of the field in bytes, and their values are operating system and media specific.
10. RODE-RECOVERED-ERROR is an RP66 status byte that indicates that an error was recovered during the data input operation. It allows the recording of error recovery activity on the input media for media monitoring, it is not an error condition and normal data will be present.
11. The RODE-SEQUENTIAL-FILE-NUMBER is initialised at 1 for each frame set and is incremented by 1 after each RODE-END-OF-FILE is written. The RODE-SEQUENTIAL-FILE-NUMBER of the RODE-END-OF-FILE IFLR is that of the preceding encapsulated file.
12. RODE-SEQUENTIAL-RECORD-NUMBER-IN-FILE is the sequential record number in the original format file. The RODE-SEQUENTIAL-RECORD-NUMBER-IN-FILE is set to 1 at the start of each encapsulated file and incremented by 1 for each encapsulated data record. The value recorded with the RODE-END-OF-FILE is the same as that recorded for the last record with a non-zero length RODE-DATA channel and is the number of records encapsulated for the encapsulated file.
13. RODE-SEISMIC-HEADER contains a variable length seismic trace header whose length is independent of that of the associated trace.
14. RODE-SEISMIC-TRACE is used for recording a seismic trace whose header information is stored in the RODE-SEISMIC-HEADER channel in the same data frame. If the input data consists of seismic traces with attached headers it should be written into the RODE-DATA channel.

The seismic trace and header construct represent an extension of simple encapsulation for specific discipline that will allow RODE to be used as working intermediate data format for modern seismic systems. These specific object names were included in the early drafts of this standard before the 0/CHANNEL:KIND attribute was specified, this construct can be extended to include as many data channels as required to define the data structure.

2.7 RODE-CONTEXT

The RODE-CONTEXT object contains metadata that describes the original source of the data and the environment that was used to create the RP66 file. The use of standard names for certain attributes facilitates decoding of the original format.

2.7.1 RODE-CONTEXT:INTERNAL-FORMAT

Currently defined standard format names are shown in Table 9

Table 9 RODE-CONTEXT:INTERNAL-FORMAT reference values

Reference value	**Description**
SEGA	seismic field tape format
SEGB	seismic field tape format
SEGC	seismic field tape format
SEGD	seismic field tape format
SEG2	seismic field tape format
SEGP	seismic location format
SEGX	seismic exchange format
SEGY	seismic exchange format
UKOOA	seismic location format
WGC4	seismic processing format
SIPMAP	seismic processing format
DISCO	seismic processing format

There are many other processing formats and more unusual field and positional formats in common use. Standard names for these formats will be established by the committee as necessary.

2.7.2 RODE-CONTEXT:SOURCE-MEDIA

The use of standard media names is essential for the management of data stored over many media types. Legacy applications also need to be able to identify the input media to allow them to select the appropriate input software. The same reference values should be used for both the RODE-CONTEXT:SOURCE-MEDIA attribute and the RODE-DIRECTORY-ENTRY:MEDIA-TYPE. Table 11 contains the list of current values.

2.8 RODE-DIRECTORY-SCOPE

2.8.1 RODE-DIRECTORY-SCOPE:DIRECTORY-SCOPE

Currently defined values correspond to simple storage management groupings are shown in Table 10.

Table 10 RODE-DIRECTORY-SCOPE:DIRECTORY-SCOPE reference values

Reference Value	**Description**
FILE-SET-DIRECTORY	Indexes all files in a file-set
STORAGE-UNIT-DIRECTORY	Indexes all files in a storage unit
STORAGE-SET-DIRECTORY	Indexes all files in a storage set
PROCESSING-HISTORY-DIRECTORY	Indexes all files used to create this file

These values should be used where appropriate - the DIRECTORY-SCOPE attribute is not restricted to the dictionary values.

2.8.2 RODE-DIRECTORY-ENTRY:MEDIA-TYPE

An initial set of reference values for the RODE-CONTEXT:SOURCE-MEDIA attribute and the RODE-DIRECTORY-ENTRY:MEDIA-TYPE are shown in Table 11. This table will be extended to include new media types in subsequent versions of the RODE dictionary.

Table 11 RODE-CONTEXT:SOURCE-MEDIA and RODE-DIRECTORY-ENTRY:MEDIA-TYPE reference values

Reference Value	**Description**
800-BPI	Industry (IBM) standard, 9 track 1/2" open reel tape, 800 BPI NRZI
1600-BPI	Industry (IBM) standard, 9 track 1/2" open reel tape, 1600 BPI PE
6250-BPI	Industry (IBM) standard, 9 track 1/2" open reel tape, 6250 BPI GCR
3480	Industry (IBM) standard 18 track 1/2" cartridge tape
3480-E	Extended length 3480 18 track 1/2" cartridge tape
3490-E	Extended length 3490 36 track 1/2" cartridge tape
DDS-4MM	4mm Helical scan DAT cartridges
EXABYTE-8200	8 mm Helical scan - 2.5 GB
EXABYTE-8500	8 mm Helical scan - 5.0 GB
DD1	Sony proprietary file format recorded on ID1
DD2	AMPEX digital implementation of 19 mm helical scan D2 cartridges.
D3	1/2" Helical scan – Storage TeK Redwood.
ST-120	Metrum T-120
TERRABYTE	CREO optical tape system.
QIC-80	Quarter inch serpentine tape
QIC-120	Quarter inch serpentine tape
QIC-150	Quarter inch serpentine tape
QIC-525	Quarter inch serpentine tape
QIC-1000	Quarter inch serpentine tape
QIC-1350	Quarter inch serpentine tape
TZ30	DEC cartridge
TK50	DEC cartridge
TK70	DEC cartridge
TZ85	DEC cartridge
TZ86	DEC cartridge
TZ87	DEC cartridge
DOS-DD	MS-DOS low density disks (720 KB)
DOS-HD	MS-DOS high density disks (1.4 MB)
MAC-DD	MAC Low density disks
MAC-HD	MAC High density disks
CD-ROM	Compact Disk ROM
LMSI1200	1.0 GB 12” Optical Disk
LMSI1500	5.6 GB 12” Optical Disk

RECORD ORIENTED DATA ENCAPSULATION FORMAT STANDARD

3. SEGY Example

3.1 Introduction

This standard contains introductory, normative and annotative information. The introductory clauses describe how the standard is organised, and provide background information on the development of this and related standards. Normative information defines the requirements that must be satisfied by any data conforming to this standard. Annotative information contains the rationale for and illustration of the normative information.

The standard is completely stated by the normative information. This example contains annotative information only and is an illustration of one way of using RP66 and RODE to encapsulate data from an old tape.

The original example published in Geophysics Vol. 61 related to a provisional version of the RP66 standard and also mis-defined the structure of the indirectly formatted logical record containing data. This error dated back to the very first examples produced by the group based on a description of the explicitly sized channel.

There are several implementations of the draft standards that used this erroneous specification of the size of the data record. Fortunately this error is immediately obvious when the data are read and can be simply corrected. A separate note has been produced for The Leading Edge and the SEG world wide web site.

3.2 Acknowledgements.

This schema is based on proposals developed by the ad-hoc formats and standards committee formed at the Seis-Media meeting held at BPX Stockley Park in April 1993.
The committee thank BP and Mobil for the use of their video-conference facilities.
The chairman thanks BP, Mobil and Statoil for their support and funding to complete the work on this schema.

3.3 Scope

This appendix contains an example of the objects and attributes that could be used to encapsulate two typical old pre-stack SEGY datasets stored on a single 1/2” tape. The reader is assumed to be familiar with the API RP66 standard and the RODE schema.

3.4 References

1. American Petroleum Institute, The October 1994 Committee draft of API Recommended Practice 66, V2, Second Edition, XXXX, 1994 "Recommended Practice for Exploration & Production Data Digital Interchange", catalog number G66002
   To order the RP66 standard contact:
   American Petroleum Institute
   Publication Department
   1220 L Street NW
   Washington DC, District of Columbia 20005
   Telephone: (202) 682-8375
   Fax: (202) 682-8537

2. Booth, Eric et al, RODE standard for record oriented data encapsulation. Geophysics Vol. 61, pp 1545-1558

3.5 Authority

This example was prepared by the Society of Exploration Geophysicists Technical Standards sub-committee for high-density media formats.

3.6 SEGY Encapsulation

3.6.1 Conventions

Attribute names are listed for each template with a count and RP66 representation code where applicable. Object names are shown as origin number, copy number, object name. They are shown in uppercase bold text.

The attribute names are right justified italics for each value of the object. They are shown for the reader's convenience and would not be part of the data written to the logical file. The characteristics of an attribute may be defined at the object level, this is shown in lower case before the values (e.g. r=ULONG defines the representation code as unsigned long for this value).

3.6.2 Input data

The input data for this example consists of two SEGY files of 3 fold CDP data stored on a 1600 BPI tape with a sticky label as shown in Figure 3. Table 12 shows the data structure of the SEGY data on tape.

Table 12 Contents of input data set

Number of records	Contents
1	3200 byte SEGY EBCDIC header
1	400 byte SEGY binary header
3366	7240 byte SEGY seismic traces
1	EOF
1	3200 byte SEGY EBCDIC header
1	400 byte SEGY binary header
3648	7240 byte SEGY seismic traces
2	EOF

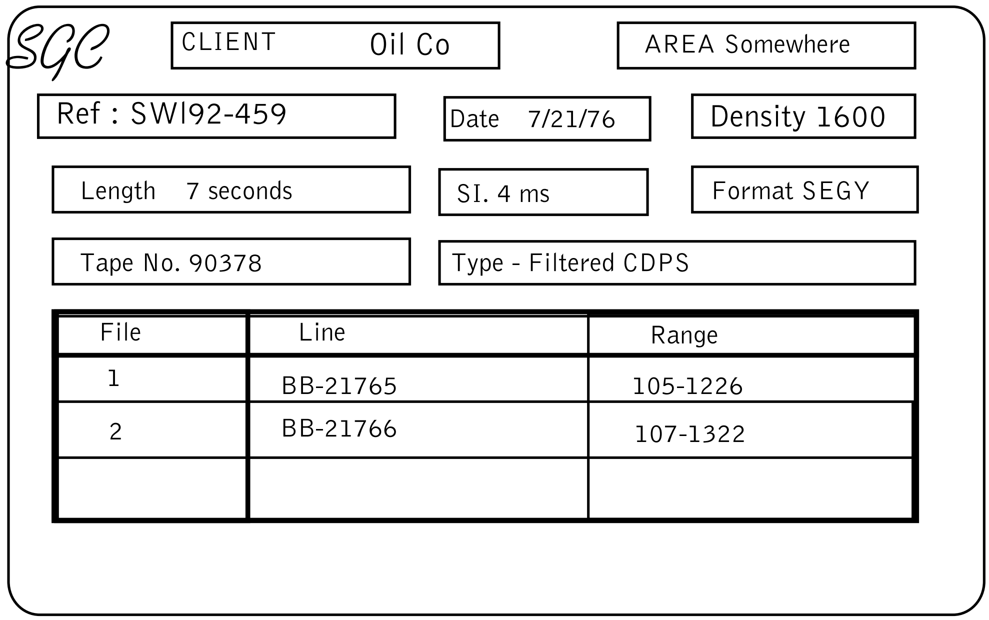


Figure 3 Typical Sticky Label

3.6.3 Objects used for RODE encapsulation of two SEGY files

3.6.3.1 Overview

This is an example of the sets of objects needed to encapsulate two SEGY files from a tape. The method chosen demonstrates several features of RP66 and RODE. RP66 is very flexible so there are alternative ways of encapsulating the files. For example, the two SEGY files could be encapsulated as a single data stream within an RP66 file (i.e. using only one frame object) or they could be encapsulated as separate logical files.

The two numbers preceding the object names are the origin number and the origin copy number. These are defined by the producer when the origin object is created and must be unique within the file.

An explicitly formatted logical record (EFLR) contains a set of objects. The set type defines the type of object for the EFLR, it includes a reference back to the defining schema. There is a template that defines the order of the attributes for each object and can contain definitions of representation codes, counts, units, default values. The objects each have a name and the values associated with the attributes in the same order as the template.

An indirectly formatted logical record contains data that can be analysed using one of the objects defined in an EFLR record (i.e. in this case the FRAME object).

The SEGY data is written into frame IFLRs. The logical records are packed by the media layer of RP66 into visible records, logical records can span visible record boundaries and hence make efficient use of the media. Table 13 shows the logical records and objects used in this example.

Table 13 RP66 Objects and data frames

Object type	Object name	Logical Record	Comments
FILE-HEADER	FILE-HEADER	1	Required by RP66
ORIGIN	API-ORIGIN	2	Defines the base schema
ORIGIN	RODE-ORIGIN	2	Defines the RODE schema

RODE-CONTEXT	RODE-CONTEXT	3	Contains RP66 metadata - who, what, when, where ....
ANCILLARY-INFORMATION	AI1 to AI15	4	Contains information from sticky label
CHANNEL	RODE-ERROR	5	Defines a channel for input tape errors
CHANNEL	RODE-END-OF-FILE	5	Defines a channel for end of file marks
CHANNEL	RODE-END-OF-TAPE	5	Defines a channel for end of tape marks
CHANNEL	RODE-DATA	5	Defines a channel for the input data
FRAME	BB-21765	6	Defines the order of channels for the encapsulated data from the first SEGY file
FRAME	BB-21766	6	Provides the order of channels for the second SEGY file.
		7-7,124	IFLRs containing data and status
RODE-SUMMARY	RODE-SUMMARY	7,125	Contains a summary of the encapsulation process

The RP66 logical file contains seven EFLRs that store metadata and control information. Each IFLR contains a single encapsulated event (i.e. either a record or an end of file mark) together with the error status from the input tape for that event plus RP66 control information.

3.6.3.2 FILE-HEADER Object.

The file header set must be the first RP66 record in a logical file, an example is shown in Table 14

Table 14 FILE-HEADER example.

Template

Attribute name	Count	Representation code
SEQUENCE-NUMBER	1	ULONG
ID	1	ASCII

Objects

Object name / *Attribute name*	**Attribute values**
698653,1,FILE-HEADER	
SEQUENCE-NUMBER	1
ID	SEGY example for RODE containing stacked datasets for lines BB21765 and BB21766

Notes:

1. The SEQUENCE-NUMBER is a positive integer, it must increase as files are written to a storage set.
2. The ID is the descriptive identifier of the file.

3.6.3.3 ORIGIN Objects

The second record in the file must contain a set of Origin objects that define how the file was created and provides the links to Schema and Dictionaries needed to interpret the file. Table 15 shows an example of the minimum ORIGIN set required for a RODE RP66 file.

Table 15 ORIGIN example

Template

Attribute name	**Count**	**Representation code**
FILE-ID	1	ASCII
FILE-SET-NAME	1	IDENT
FILE-SET-NUMBER	1	ULONG
FILE-NUMBER	1	ULONG
FILE-TYPE	1	IDENT
CREATION-TIME	1	DTIME
SCHEMA-CODE	1	ULONG
SCHEMA-ORGANISATION	1	ASCII
SCHEMA-EDITION	1	IDENT
SCHEMA-DICTIONARY-EDITION	1	IDENT
CONTEXT	1	OBJREF
NAMESPACE-CODE	1	ULONG
NAMESPACE-NAME	1	IDENT
NAMESPACE-ORGANISATION	1	ASCII
NAMESPACE-EDITION	1	IDENT

Objects

Object name / *Attribute name*	**Attribute values**
698653,1,API-ORIGIN	
FILE-ID	SEGY example for RODE containing stacked datasets for lines BB21765 and BB21766
FILE-SET-NAME	SEG.RODE.EXAMPLES
FILE-SET-NUMBER	1
FILE-NUMBER	1
FILE-TYPE	FILTERED-CDPS
CREATION-TIME	1-DEC-1996
SCHEMA-CODE	0
SCHEMA-ORGANISATION	API Subcommittee On Recommended Format for Digital Well Data, Basic Schema
SCHEMA-EDITION	1
SCHEMA-DICTIONARY-EDITION	1
CONTEXT	*Absent*
NAMESPACE-CODE	463
NAMESPACE-NAME	RODE
NAMESPACE-ORGANISATION	SEG High Density Media Format Sub-committee
NAMESPACE-EDITION	1.01
599357,1,RODE-ORIGIN	
FILE-ID	SEGY example for RODE containing stacked datasets for lines BB21765 and BB21766
FILE-SET-NAME	SEG.RODE.EXAMPLES
FILE-SET-NUMBER	1
FILE-NUMBER	1
FILE-TYPE	FILTERED-CDPS
CREATION-TIME	1-DEC-1996
SCHEMA-CODE	463
SCHEMA-ORGANISATION	SEG High Density Media Format Sub-committee
SCHEMA-EDITION	1.01
SCHEMA-DICTIONARY-EDITION	1.01
CONTEXT	599357,1,RODE-CONTEXT
NAMESPACE-CODE	463
NAMESPACE-NAME	RODE
NAMESPACE-ORGANISATION	SEG High Density Media Format Sub-committee
NAMESPACE-EDITION	1.01

Notes:

The origin and copy numbers defined here are used by all other objects in the file to select the origin that is used to define the use of the object and its attributes. The API-ORIGIN object is the defining origin for this file. The Origin numbers are generated using a random number generator to allow data from many RP66 files to be concatenated into a single file.

3.6.3.4 RODE CONTEXT Object

The RODE-CONTEXT object contains RODE specific meta data. Table 16 shows an example of a RODE-CONTEXT logical record, the META-DATA-ROOT attribute is not used in this example and is not present in the template.

Table 16 RODE-CONTEXT example

Template

Attribute name	Count	Representation code
INTERNAL-FORMAT	1	IDENT
OPERATING-SYSTEM	1	ASCII
PROGRAM		ASCII
PROGRAM-VERSION		ASCII
AUTHOR-CODE	1	ULONG
AUTHOR-NAME	1	ASCII
PRODUCER-CODE	1	ULONG
PRODUCER-NAME	1	ASCII
PRODUCER-SYSTEM	1	ASCII
PRODUCER-DRIVE	1	ASCII
CLIENT-COMPANY		ASCII
SOURCE-COMPANY	1	ASCII
SOURCE-PRODUCER-CODE	1	ULONG
SOURCE-COMMENTS	1	ASCII
SOURCE-CREATION-TIME	1	DTIME
SOURCE-IDENTIFIER	1	ASCII
SOURCE-MEDIA	1	IDENT

Objects

Object name / *Attribute name*	**Attribute values**
599357,1,RODE-CONTEXT	
INTERNAL-FORMAT	SEGY
OPERATING-SYSTEM	SOME OS-VN.M
PROGRAM	Runrode
PROGRAM-VERSION	V1.1
AUTHOR-CODE	*"Absent attribute"*
AUTHOR-NAME	Author of the RODE encoding software
PRODUCER-CODE	999
PRODUCER-NAME	Contractor
PRODUCER-SYSTEM	Copycat
PRODUCER-DRIVE	/dev/rmt4
CLIENT-COMPANY	Any Oil Company
SOURCE-COMPANY	Original processing contractor
SOURCE-PRODUCER-CODE	*"Absent attribute"*
SOURCE-COMMENTS	General comments - e.g. purchased with lease ....
SOURCE-CREATION-TIME	76/7/21 00:00:00.000
SOURCE-IDENTIFIER	B01001
SOURCE-MEDIA	1600-BPI

3.6.3.5 ANCILLARY-INFORMATION Object

Figure 3 shows an example of a tape label that is encoded in the ANCILLARY-INFORMATION record in Table 17.

Table 17 ANCILLARY-INFORMATION example

Template

Attribute name	Count	Representation code
DESIGNATION	1	ASCII
VALUE	1	ASCII

Objects

Object name / *Attribute name*	**Attribute values**		
599357,1,AI1			
DESIGNATION	Contractor		
VALUE	SGC		
599357,1,AI2			
DESIGNATION	Client		
VALUE	Oil Co.		
599357,1,AI3			
DESIGNATION	AREA		
VALUE	Somewhere		
599357,1,AI4			
DESIGNATION	Ref.		
VALUE	SW192-459		
599357,1,AI5			
DESIGNATION	Date		
VALUE	r=dtime,v=76/7/21 00:00:00.000		
599357,1,AI6			
DESIGNATION	Density		
VALUE	1600		
599357,1,AI7			
DESIGNATION	Length		
VALUE	r=ULONG,u=s,v=7		
599357,1,AI8			
DESIGNATION	Sample Interval		
VALUE	r=ULONG,u=ms,v=4		
599357,1,AI9			
DESIGNATION	Format		
VALUE	r=IDENT,v=SEGY		
599357,1,AI10			
DESIGNATION	Tape No.		
VALUE	r=IDENT,v=90378		
599357,1,AI11			
DESIGNATION	Type		
VALUE	Filtered CDPS		
599357,1,AI12			
DESIGNATION	File		
VALUE	c=2,r=UNORM,v=	1	2
599357,1,AI13			
DESIGNATION	Line		
VALUE	c=2,r=IDENT,v=	BB-21765	BB-21766
599357,1,AI14			
DESIGNATION	First ensemble		
VALUE	c=2,r=ULONG,v=	105	107
599357,1,AI15			
DESIGNATION	Last Ensemble		
VALUE	c=2,r=ULONG,v=	1226	1322

N.B. RODE does not provide a default for the count or representation code for VALUE field of an ANCILLARY-INFORMATION object. These fields are independently specified for each object, unless units are specified they are assumed to be irrelevant. A default representation code of ASCII is specified for the VALUE field, this can still be overridden at the individual attribute value level.

3.6.3.6 CHANNEL Objects

Table 18 contains an example of the channel logical record that defines the channels that may be recorded in this encapsulation run.

Table 18 CHANNEL example

Template

Attribute name	**Count**	**Representation code**
DESCRIPTION	1	ASCII
FLAGS		IDENT
REPRESENTATION-CODE	1	USHORT
DIMENSION	1	ULONG
DIMENSION-LIMIT	1	ULONG
ELEMENT-LIMIT	1	ULONG

Objects

Object name / *Attribute name*	**Attribute values**	**Notes**
698653,1,RODE-ERROR		
DESCRIPTION	RODE input tape status	
FLAGS	*Absent*	
REPRESENTATION-CODE	26	
DIMENSION	1	1
698653,1,RODE-END-OF-FILE		
DESCRIPTION	RODE input tape end of file	
FLAGS	*absent*	
REPRESENTATION-CODE	26	
DIMENSION	1	1
698653,1,RODE-END-OF-TAPE		
DESCRIPTION	Rode input tape end of media warning	
FLAGS	*absent*	
REPRESENTATION-CODE	26	
DIMENSION	1	1
698653,2,RODE-DATA		3
DESCRIPTION	Rode input tape data	
FLAGS	EXPLICIT-SIZE	
REPRESENTATION-CODE	15	
DIMENSION	*absent*	*2*
DIMENSION-LIMIT	1	
ELEMENT-LIMIT	16,128	3
698653,2,RODE-ERROR		4
DESCRIPTION	RODE input tape status	
FLAGS	*absent*	
REPRESENTATION-CODE	26	
DIMENSION	1	1
698653,2,RODE-END-OF-FILE		4
DESCRIPTION	RODE input tape end of file	
FLAGS	*absent*	
REPRESENTATION-CODE	26	
DIMENSION	1	1
698653,2,RODE-END-OF-TAPE		4
DESCRIPTION	Rode input tape end of media warning	
FLAGS	*absent*	
REPRESENTATION-CODE	26	
DIMENSION	1	1
698653,2,RODE-DATA		4
DESCRIPTION	Rode input tape data	
FLAGS	EXPLICIT-SIZE	
REPRESENTATION-CODE	15	
DIMENSION	*absent*	*2*
DIMENSION-LIMIT	1	
ELEMENT-LIMIT	16,128	3

Notes:

1. Attribute values at the end of the template list are assumed to be absent if another object name is encountered. The DIMENSION-LIMIT and ELEMENT-LIMIT attributes are only meaningful for an explicitly sized channel.
2. The DIMENSION-LIMIT will be 1 for all RODE datasets, it is required for explicitly sized channels and specifies the structure of the channel.
3. ELEMENT-LIMIT is set to 16 KB, a value is required fro explicitly sized channels and could be used to define the application program's I/O buffer size.
4. A channel object represents a single instance of a sequence of values in a frame type. In this example we are using different frame objects for each data set on tape and therefore need two sets of channel objects. The second set of objects are copies of the first set and differ only by copy number.

3.6.3.7 FRAME Object

Table 13 contains two frame objects that define which channels are used and their order within each frame of the data.

Table 19 FRAME object example

Template

Attribute name	**Count**	**Representation code**
DESCRIPTION	1	ASCII
CHANNELS		OBNAME
FRAMES-PER-IFLR-LIMIT	1	ULONG

Objects

Object name / ***Attribute name***	**Attribute values**
698653,1,BB-21765	
DESCRIPTION	SEGY encapsulation for line BB-21765
CHANNELS	c=4,698653,1,RODE-ERROR 698653,1,RODE-END-OF-FILE 698653,1,RODE-END-OF-TAPE 698653,1,RODE-DATA
FRAMES-PER-IFLR-LIMIT	1
698653,1,BB-21766	
DESCRIPTION	SEGY encapsulation for line BB-21766
CHANNELS	c=4, 698653,2,RODE-ERROR 698653,2,RODE-END-OF-FILE 698653,2,RODE-END-OF-TAPE 698653,2,RODE-DATA
FRAMES-PER-IFLR-LIMIT	1

Notes:

1. The user does not need to know the order in which the SEGY files are written in the logical file. A specific SEGY dataset can be selected by choosing the appropriate data frames.
2. The FRAMES-PER-IFLR-LIMIT is set to 1, the IFLR overhead is very small compared with most geophysical record lengths and this makes for much simpler IFLR records.

3.6.3.8 Frame Data IFLR

Once all the set up logical records have been created, the data can be recorded in frame data IFLRs , each input read operation will generate a single output logical record as shown in Table 20.

Table 20 FRAME IFLR structure

Bytes	Contents	Notes
1-6	Logical Record Segment Header	The attribute byte indicates this is an IFLR
7-20	698653,1,BB-21765	The data descriptor is the frame object 698653,1,BB-21765
21	0	IFLR modifier - data
22-25	1	number of frames
26-29	Frame Number	Increment for each frame in the FRAME object data set
30-33	1	Dimension count
34-37	Number of bytes in input record.	length of input data
38	Input Tape Error status	
39	Input Tape End of File	
40	Input Tape - End of Tape detected.	
41-n	Input data	
Padding and pad count.		If required.

Once all the input data for the first SEGY file has been read (i.e. the first EOF is encountered on the input media and copied to the encapsulated data) an EOD IFLR should be written as follows:

Table 21 EOD IFLR example

Bytes	Contents	Notes
1-6	Logical Record Segment Header	The attribute byte indicates this is an IFLR
7-20	698653,1,BB-21765	The data descriptor is the frame object 698653,1,BB-21765
21	1	IFLR modifier indicating that this is an End of Data record.
22-28	Padding and pad count rounding to full 4 byte words.	The minimum Logical Record Segment is 16 bytes.

The second SEGY file is then encapsulated as frame IFLRs with a data descriptor reference of 698653,1,BB-21766. The sequence of IFLRs will be completed by one containing the EOF, an EOD would be written.

3.6.3.9 RODE-SUMMARY Object

When the second EOF is read on the input (i.e. a double EOF or a zero length file used by most geophysical programs to indicate the end of data on a tape), the encapsulation program would write the summary object as shown in Table 22.

Table 22 RODE-SUMMARY example

Template

Attribute name	Count	Representation code
NUMBER-OF-BYTES-LOW	1	ULONG
NUMBER-OF-RECORDS	1	ULONG
NUMBER-OF-FILES	1	ULONG
NUMBER-OF-ERRORS	1	ULONG
NUMBER-OF-EOTS	1	ULONG
NUMBER-OF-EOFS	1	ULONG

Objects

Object name / ***Attribute name***	**Attribute values**
599357,1,RODE-SUMMARY	
NUMBER-OF-BYTES-LOW	50,788,560
NUMBER-OF-RECORDS	7,018
NUMBER-OF-FILES	2
NUMBER-OF-ERRORS	0
NUMBER-OF-EOTS	0
NUMBER-OF-EOFS	2

Notes:

1. The total number of bytes read is 50,788,560- the sum of the bytes of the lengths of SEGY header records and SEGY traces.
2. A total of 2,442 records were stored, 2 headers and 3,366 traces from the first file, and 2 headers and 3,648 traces from the second file.
3. There were two input files.
4. There were two input end of file marks, one between the two input files and one at the end of the dataset. The final end-of-file mark was not transferred to the output media and hence is not counted in the summary.