The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

LOSSLESS DATA COMPRESSION

RECOMMENDED STANDARD

CCSDS 121.0-B-3

BLUE BOOK

August 2020

The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

LOSSLESS DATA COMPRESSION

RECOMMENDED STANDARD

CCSDS 121.0-B-3

BLUE BOOK

August 2020

DEDICATION

This document is dedicated to the memory of Mr. Warner H. Miller of NASA. Warner had been with the CCSDS since its beginning, and throughout the years he was a major contributor to numerous standards for error control coding, radio frequency modulation, and data architecture. He initiated this data compression standard and saw its publication and use by many space missions. Warner was a superb technologist, a gentleman, and a friend always ready to help, especially young colleagues. Warner and his approach to work and life in general will be deeply missed by his many friends and colleagues in the CCSDS.

AUTHORITY

Issue:	Recommended Standard, Issue 3
Date:	August 2020
Location:	Washington, DC, USA

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and represents the consensus technical agreement of the participating CCSDS Member Agencies. The procedure for review and authorization of CCSDS documents is detailed in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4), and the record of Agency participation in the authorization of this document can be obtained from the CCSDS Secretariat at the e-mail address below.

This document is published and maintained by:

CCSDS Secretariat
National Aeronautics and Space Administration
Washington, DC, USA
Email: secretariat@mailman.ccsds.org

STATEMENT OF INTENT

The Consultative Committee for Space Data Systems (CCSDS) is an organization officially established by the management of its members. The Committee meets periodically to address data systems problems that are common to all participants, and to formulate sound technical solutions to these problems. Inasmuch as participation in the CCSDS is completely voluntary, the results of Committee actions are termed **Recommended Standards** and are not considered binding on any Agency.

This **Recommended Standard** is issued by, and represents the consensus of, the CCSDS members. Endorsement of this **Recommendation** is entirely voluntary. Endorsement, however, indicates the following understandings:

- o Whenever a member establishes a CCSDS-related **standard**, this **standard** will be in accord with the relevant **Recommended Standard**. Establishing such a **standard** does not preclude other provisions which a member may develop.

- o Whenever a member establishes a CCSDS-related **standard**, that member will provide other CCSDS members with the following information:

  - -- The **standard** itself.

  - -- The anticipated date of initial operational capability.

  - -- The anticipated duration of operational service.

- o Specific service arrangements shall be made via memoranda of agreement. Neither this **Recommended Standard** nor any ensuing **standard** is a substitute for a memorandum of agreement.

No later than five years from its date of issuance, this **Recommended Standard** will be reviewed by the CCSDS to determine whether it should: (1) remain in effect without change; (2) be changed to reflect the impact of new technologies, new requirements, or new directions; or (3) be retired or canceled.

In those instances when a new version of a **Recommended Standard** is issued, existing CCSDS-related member standards and implementations are not negated or deemed to be non-CCSDS compatible. It is the responsibility of each member to determine when such standards or implementations are to be modified. Each member is, however, strongly encouraged to direct planning for its new standards and implementations towards the later version of the Recommended Standard.

FOREWORD

This Recommended Standard establishes a common framework and provides a common basis for a Lossless data compression algorithm applicable to several different types of data.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CCSDS has processes for identifying patent issues and for securing from the patent holder agreement that all licensing policies are reasonable and non-discriminatory. However, CCSDS does not have a patent law staff, and CCSDS shall not be held responsible for identifying any or all such patent rights.

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Recommended Standard is therefore subject to CCSDS document management and change control procedures, which are defined in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4). Current versions of CCSDS documents are maintained at the CCSDS Web site:

http://www.ccsds.org/

Questions relating to the contents or status of this document should be sent to the CCSDS Secretariat at the email address indicated on page i.

At time of publication, the active Member and Observer Agencies of the CCSDS were:

Member Agencies

- Agenzia Spaziale Italiana (ASI)/Italy.
- Canadian Space Agency (CSA)/Canada.
- Centre National d'Etudes Spatiales (CNES)/France.
- China National Space Administration (CNSA)/People's Republic of China.
- Deutsches Zentrum für Luft- und Raumfahrt (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Federal Space Agency (FSA)/Russian Federation.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- Japan Aerospace Exploration Agency (JAXA)/Japan.
- National Aeronautics and Space Administration (NASA)/USA.
- UK Space Agency/United Kingdom.

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Belgian Federal Science Policy Office (BFSPO)/Belgium.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- China Satellite Launch and Tracking Control General, Beijing Institute of Tracking and Telecommunications Technology (CLTC/BITTT)/China.
- Chinese Academy of Sciences (CAS)/China.
- China Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- Danish National Space Center (DNSC)/Denmark.
- Departamento de Ciência e Tecnologia Aeroespacial (DCTA)/Brazil.
- Electronics and Telecommunications Research Institute (ETRI)/Korea.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Geo-Informatics and Space Technology Development Agency (GISTDA)/Thailand.
- Hellenic National Space Committee (HNSC)/Greece.
- Hellenic Space Agency (HSA)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space Research (IKI)/Russian Federation.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- Mohammed Bin Rashid Space Centre (MBRSC)/United Arab Emirates.
- National Institute of Information and Communications Technology (NICT)/Japan.
- National Oceanic and Atmospheric Administration (NOAA)/USA.
- National Space Agency of the Republic of Kazakhstan (NSARK)/Kazakhstan.
- National Space Organization (NSPO)/Chinese Taipei.
- Naval Center for Space Technology (NCST)/USA.
- Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- Scientific and Technological Research Council of Turkey (TUBITAK)/Turkey.
- South African National Space Agency (SANSA)/Republic of South Africa.
- Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.
- Swiss Space Office (SSO)/Switzerland.
- United States Geological Survey (USGS)/USA.

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 121.0-B-1	Lossless Data Compression, Blue Book, Issue 1	May 1997	Original issue, superseded
CCSDS 121.0-B-2	Lossless Data Compression, Recommended Standard, Issue 2	May 2012	Issue 2, superseded
CCSDS 121.0-B-3	Lossless Data Compression, Recommended Standard, Issue 3	August 2020	Current issue (note)

NOTE – Textual changes in the current issue are too numerous to permit meaningful application of change bars. Changes from the previous issue are enumerated as follows:

Changes affecting backward compatibility:

– explicitly defines prediction value of first sample;

– explicitly defines mapped value δ_1=0 for second-extension and zero-block options when including reference samples;

– specifies that at the end of the input sequence less than r blocks may follow the last reference sample;

– changes definition of reference sample interval, *r*, from 'CDSes between reference samples' to 'blocks between reference samples';

– specifies the size of a segment as 64 blocks, except possibly the last of the reference sample interval or the last of the sequence, which may be smaller;

– specifies that ROS shall be also used in segments smaller than 64 blocks (i.e., end of reference sample interval or end of input sequence);

– specifies the values to be used in the preprocessor subfield of the CIP source data field (source configuration field) when the preprocessor is absent.

Changes not affecting backward compatibility:

– clarifies that a CDS can encode more than one block of samples when sequences of zero blocks are found;

– clarifies use of bypass prediction or bypass preprocessor;

– clarifies insertion of reference samples and mapped/predicted samples into the CDS;

– adds new section defining optional file format.

CONTENTS

Section Page

1 INTRODUCTION 1-1

1.1 PURPOSE 1-1
1.2 SCOPE 1-1
1.3 APPLICABILITY 1-1
1.4 RATIONALE 1-1
1.5 CONVENTIONS AND DEFINITIONS 1-2
1.6 REFERENCES 1-2

2 OVERVIEW 2-1

2.1 GENERAL 2-1
2.2 THE SOURCE CODER 2-1
2.3 DATA TRANSMISSION 2-3

3 ADAPTIVE ENTROPY CODER 3-1

3.1 CODE SPECIFICATION 3-1
3.2 THE FUNDAMENTAL SEQUENCE OPTION 3-2
3.3 SPLIT-SAMPLE OPTIONS 3-3
3.4 THE SECOND-EXTENSION OPTION 3-4
3.5 THE ZERO-BLOCK OPTION 3-4
3.6 THE NO-COMPRESSION OPTION 3-5
3.7 CODE SELECTION 3-5

4 PREPROCESSOR 4-1

4.1 GENERAL PREPROCESSOR FUNCTION 4-1
4.2 PREDICTORS 4-1
4.3 REFERENCE SAMPLE INTERVAL 4-3
4.4 PREDICTION ERROR MAPPER 4-3

5 CODED DATA SET AND PACKET FORMATS 5-1

5.1 OVERVIEW 5-1
5.2 THE CODED DATA SET FORMAT 5-1
5.3 CODED DATA SETS INSERTED INTO SPACE PACKETS 5-5

CONTENTS (continued)

Section Page

6 COMPRESSION IDENTIFICATION PACKET (OPTIONAL) 6-1

6.1 COMPRESSION IDENTIFICATION PACKET STRUCTURE 6-1
6.2 CIP PACKET PRIMARY HEADER 6-1
6.3 PACKET DATA FIELD 6-2

7 FILE FORMAT 7-1

7.1 OVERVIEW 7-1
7.2 FILE STRUCTURE 7-1

ANNEX A SECURITY, SANA, AND PATENT CONSIDERATIONS (INFORMATIVE) A-1
ANNEX B GLOSSARY OF ACRONYMS AND TERMS (INFORMATIVE) B-1
ANNEX C INFORMATIVE REFERENCES (INFORMATIVE) C-1

Figure

2-1 Schematic of the Source Coder 2-1
3-1 The Adaptive Entropy Coder with a Preprocessor 3-1
3-2 Split-Sample Format 3-3
4-1 A Preprocessor 4-1
4-2 Preprocessor Using a Unit-Delay Predictor 4-2
5-1 CDS Format When a Split-Sample Option Is Selected (Including the Special Case k=0 Equivalent to the FS Option) 5-3
5-2 CDS Format When the No-Compression Option Is Selected 5-3
5-3 CDS Format When the Zero-Block Option Is Selected 5-4
5-4 CDS Format When the Second-Extension Option Is Selected 5-5
5-5 Packet Format for l CDSes 5-5
6-1 CIP Structure 6-1
6-2 Subfields of the Source Configuration Field 6-3
7-1 File Format 7-1

Table

3-1 Fundamental Sequence Codewords As a Function of the Preprocessed Samples 3-3
3-2 Zero-Block Fundamental Sequence Codewords As a Function of the Number of Consecutive All-Zeros Blocks 3-5
5-1 Selected Code Option Identification Key 5-2
7-1 File Header Structure 7-1

1 INTRODUCTION

1.1 PURPOSE

The purpose of this document is to establish a Recommended Standard for a source-coding data-compression algorithm applied to digital data and to specify how these compressed data shall be inserted into space packets and files for retrieval and decoding.

Source coding for data compression is a method utilized in data systems to reduce the volume of digital data to achieve benefits in areas including, but not limited to,

a) reduction of transmission channel bandwidth;

b) reduction of the buffering and storage requirement;

c) reduction of data-transmission time at a given rate.

1.2 SCOPE

The characteristics of source codes are specified only to the extent necessary to ensure multi-mission support capabilities. The specification does not attempt to quantify the relative bandwidth reduction, the merits of each approach discussed, or the design requirements for coders and associated decoders. Some performance information is included in reference [C2].

This Recommended Standard addresses only Lossless source coding, which is applicable to a wide range of digital data, both imaging and non-imaging, in which the requirement is for a moderate data-rate reduction constrained to allow no distortion to be added in the data compression/decompression process. The decompression process is not addressed. (See reference [C2] for an outline of an implementation.)

1.3 APPLICABILITY

This Recommended Standard applies to data compression applications for space missions anticipating packetized telemetry cross support. In addition, it serves as a guideline for the development of compatible CCSDS Agency standards in this field, based on good engineering practice.

1.4 RATIONALE

The concept and rationale for the Lossless source coding for data compression algorithm described herein may be found in reference [C2].

1.5 CONVENTIONS AND DEFINITIONS

1.5.1 MATHEMATICAL NOTATIONS AND DEFINITIONS

In this document, for any real number x, the largest integer p such that $p \leq x$ is denoted by

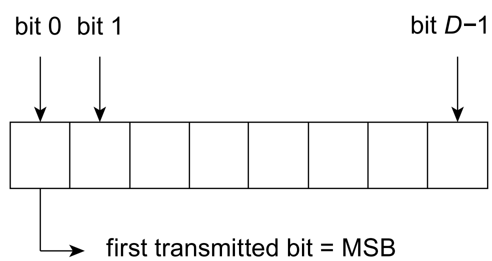
$$p = \lfloor x \rfloor$$

The modulus of an integer M with respect to a positive integer divisor p, denoted $M \bmod p$ is defined to be

$$M \bmod p = M - p\lfloor M / p \rfloor$$

1.5.2 CONVENTIONS

In this document, the following convention is used to identify each bit in a D–bit word. The first bit in the word to be transmitted (i.e., the most left justified when drawing a figure) is defined to be 'bit 0', the following bit is defined to be 'bit 1', and so on up to 'bit D–1'. When the word is used to express an unsigned binary value (such as a counter), the Most Significant Bit (MSB) shall correspond to the highest power of two, that is, 2^{D-1}.



In accordance with modern data communications practice, spacecraft data words are often grouped into 8-bit 'words' that conform to the above convention. Throughout this Recommended Standard, the following nomenclature is used to describe this grouping:

8-bit Word = 'Byte'

1.6 REFERENCES

The following documents contain provisions which, through reference in this text, constitute provisions of this Recommended Standard. At the time of publication, the editions indicated were valid. All documents are subject to revision, and users of this Recommended Standard are encouraged to investigate the possibility of applying the most recent editions of the documents indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS Recommended Standards.

[1] *TM Synchronization and Channel Coding*. Issue 3. Recommendation for Space Data System Standards (Blue Book), CCSDS 131.0-B-3. Washington, D.C.: CCSDS, September 2017.

[2] *Flexible Advanced Coding and Modulation Scheme for High Rate Telemetry Applications*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 131.2-B-1. Washington, D.C.: CCSDS, March 2012.

[3] *CCSDS Space Link Protocols over ETSI DVB-S2 Standard*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 131.3-B-1. Washington, D.C.: CCSDS, March 2013.

[4] *Space Packet Protocol*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 133.0-B-2. Washington, D.C.: CCSDS, June 2020.

[5] *Low-Complexity Lossless and Near-Lossless Multispectral and Hyperspectral Image Compression*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 123.0-B-2. Washington, D.C.: CCSDS, February 2019.

[6] *CCSDS File Delivery Protocol (CFDP)*. Issue 4. Recommendation for Space Data System Standards (Blue Book), CCSDS 727.0-B-4. Washington, D.C.: CCSDS, January 2007.

[7] *TM Space Data Link Protocol*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 132.0-B-2. Washington, D.C.: CCSDS, September 2015.

[8] *AOS Space Data Link Protocol*. Issue 3. Recommendation for Space Data System Standards (Blue Book), CCSDS 732.0-B-3. Washington, D.C.: CCSDS, September 2015.

[9] *Unified Space Data Link Protocol*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 732.1-B-1. Washington, D.C.: CCSDS, October 2018.

2 OVERVIEW

2.1 GENERAL

This Recommended Standard defines for standardization a particular adaptive source coding algorithm that has widespread applicability to many forms of digital data. In particular, the science data from many types of imaging or non-imaging instruments are well suited for the application of this algorithm.

There are two classes of source coding methods: Lossless and Lossy.

A Lossless source coding technique preserves source data accuracy and removes redundancy in the data source. In the decoding process, the original data can be reconstructed from the compressed data by restoring the removed redundancy; the decompression process adds no distortion. This technique is particularly useful when data integrity cannot be compromised. The price it pays is generally a lower Compression Ratio, which is defined as the ratio of the number of original uncompressed bits to the number of compressed bits including overhead bits necessary for signaling parameters.

On the other hand, a Lossy source coding method removes some of the source information content along with the redundancy. The original data cannot be fully restored, and data distortion occurs. However, if some distortion can be tolerated, Lossy source coding generally achieves a higher compression ratio. By controlling the amount of acceptable distortion and compression, this technique may enable acquisition and dissemination of mission data within a critical time span.

This Recommended Standard addresses only Lossless source coding and does not attempt to explain the theory underlying the operation of the algorithm.

2.2 THE SOURCE CODER

The Lossless source coder consists of two separate functional parts: the preprocessor and the adaptive entropy coder, as shown in figure 2-1.

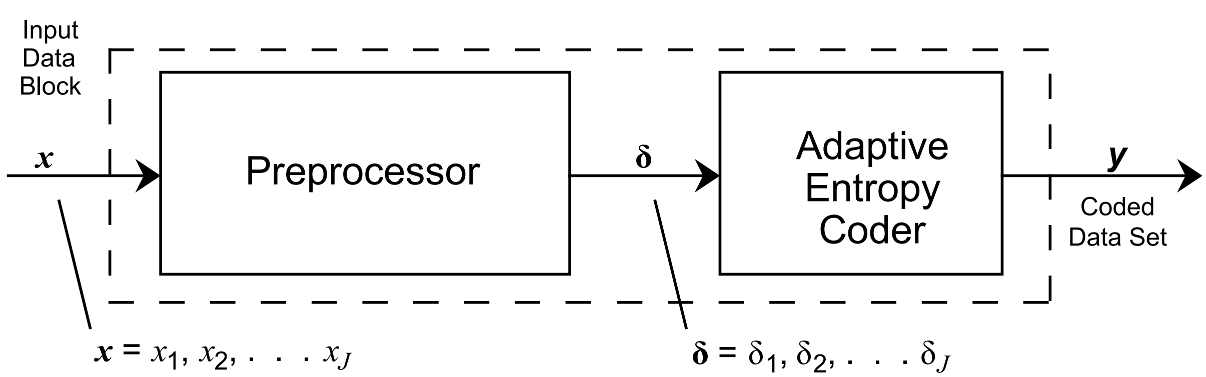


Figure 2-1: Schematic of the Source Coder

Inputs to the source coder are partitioned into blocks of J n-bit samples,

$$\boldsymbol{x} = x_1, x_2, \ldots x_J,$$

where J and n are constant values.

When the input sequence length is not a multiple of J, a user must append additional ‘padding’ samples as needed. Compressed data size will be minimized when padding samples are chosen so that the corresponding padded preprocessed samples are zero.

Preprocessor:

The preprocessor applies a reversible function to each block of input data samples $\boldsymbol{x}$, to produce a ‘preferred’ source block of the same length:

$$\boldsymbol{\delta} = \delta_1, \delta_2, \ldots \delta_i, \ldots \delta_J,$$

where each δ_i is an n-bit integer, $0 \leq \delta_i \leq (2^n–1)$. For an ideal preprocessing stage, $\boldsymbol{\delta}$ will have the following properties:

a) the $\{\delta_i\}$ is statistically independent and identically distributed;

b) the preferred probability, p_m, that any sample δ_i will take on integer value m is a nonincreasing function of value m, for $m = 0, 1, \ldots (2^n–1)$.

The preprocessor function must be a reversible operation, and, in general, the best Lossless preprocessor will meet the above conditions and produce the lowest entropy, which is a measure of the smallest average number of bits that can be used to represent each sample.

When the preprocessor is a predictor, its outputs $\boldsymbol{\delta} = \delta_1, \ldots \delta_i, \ldots \delta_J$ represent prediction errors. In this case, periodic insertion of reference samples may be required to make the transformation from input samples $\boldsymbol{x}$ to preprocessed outputs $\boldsymbol{\delta}$ reversible, and to prevent transmission channel errors from propagating excessively through the decompressed data. If reference samples are used, they are inserted periodically according to a user-specified reference sample interval r, and they must always be the first sample x_1 of a J-sample block.

If an input data block $\boldsymbol{x}$ includes a reference sample, it does not strictly follow the data flow depicted in figure 2-1. In this special case, the entire block $\boldsymbol{x} = x_1, x_2, \ldots x_J$ is input to the Preprocessor, but its output presented to the Adaptive Entropy Coder consists of the unprocessed reference sample x_1, followed by (J–1) ‘preferred’ samples $\boldsymbol{\delta} = \delta_2, \ldots \delta_i, \ldots \delta_J$. The entropy coder in this case passes the uncoded reference sample x_1 directly into the corresponding coded data set, and it applies its various coding options only to the (J–1) ‘preferred’ samples $\boldsymbol{\delta} = \delta_2, \ldots \delta_i, \ldots \delta_J$.

For the Zero-Block and the Second-Extension options, coding proceeds using $\delta_1 = 0$. This Recommended Standard does not attempt to explain methods for choosing a preprocessing stage. This Recommended Standard does provide the definition of a basic Unit-Delay Predictor preprocessing stage (see 4.2.5) that may be suitable for many applications. However, it is important that users carefully address this issue since careful selection of an appropriate preprocessing stage is essential for efficient compression and depends on the source-data characteristics. Interested users should refer to reference [C2].

Adaptive Entropy Coder:

The function of the Adaptive Entropy Coder is to calculate uniquely decipherable, variable-length codewords corresponding to each block of preprocessed samples $\boldsymbol{\delta}$. The entropy coder incorporates multiple coding options, each exhibiting efficient performance over different yet overlapping ranges of entropy. The coder selects the coding option that gives the highest compression ratio among the various options on each block of J preprocessed samples. A code-option 'identifier', requiring only a few bits, is attached before the first codeword bit in a coded block to signal the coding option to the decoder for proper decompression. Since the block size J can be small and a new code option is selected for each block, the overall coding can adapt to rapid changes in data statistics.

The variable-length encoded bit sequence output from the Adaptive Entropy Coder to represent a J-sample block is called a Coded Data Set (CDS). The formatting of CDSes is specified in section 5.

2.3 DATA TRANSMISSION

Compressed CDS data from the Adaptive Entropy Coder are inserted into space packets, groups of packets, or files for transmission. Formatting of packets, groups of packets, and files is specified in 5.3 and sections 6 and 7.

In case the encoded stream is to be transmitted over a CCSDS space link, several protocols can be used to transfer the CDSes, including but not limited to:

– Space Packet Protocol (see reference [4]);

– CCSDS File Delivery Protocol (CFDP) (see reference [6]);

– Packet service as provided by the CCSDS Space Data Link Protocols (see references [7], [8], and [9]).

Limits on the maximum size data unit that can be transmitted may be imposed by the protocol used or by other practical implementation considerations. The user is expected to take such limits into account when using this Recommended Standard.

This Recommended Standard does not incorporate sync markers or other mechanisms to flag the packets or file headers, or the beginning of a reference sample interval; it is assumed that the transport mechanism used for the delivery of the encoded bit stream will provide the ability to locate the beginning and end of the compressed data and, in the event of data corruption, the beginning of the next packet, file, or reference sample interval.

When transmission over a CCSDS space link occurs, application of one of the set of Channel Coding and Synchronization Recommended Standards (references [1], [2], and [3]) will significantly reduce the loss of portions of transmitted data caused by data corruption over the transmission channel. This is important because individual channel bit errors have greater consequences when data are compressed. The effects of a small error or data loss event can propagate to corrupt an entire compressed sequence of samples. Therefore measures should be taken to minimize errors and data loss in the compressed data.

3 ADAPTIVE ENTROPY CODER

3.1 CODE SPECIFICATION

3.1.1 Figure 3-1 represents the general-purpose Adaptive Entropy Coder with a preprocessor. Basically, such a coder chooses one of a set of code options to represent an incoming block of preprocessed data samples, **δ**. A unique identifier (ID) bit sequence is attached to the code block to indicate to the decoder which decoding option to use.

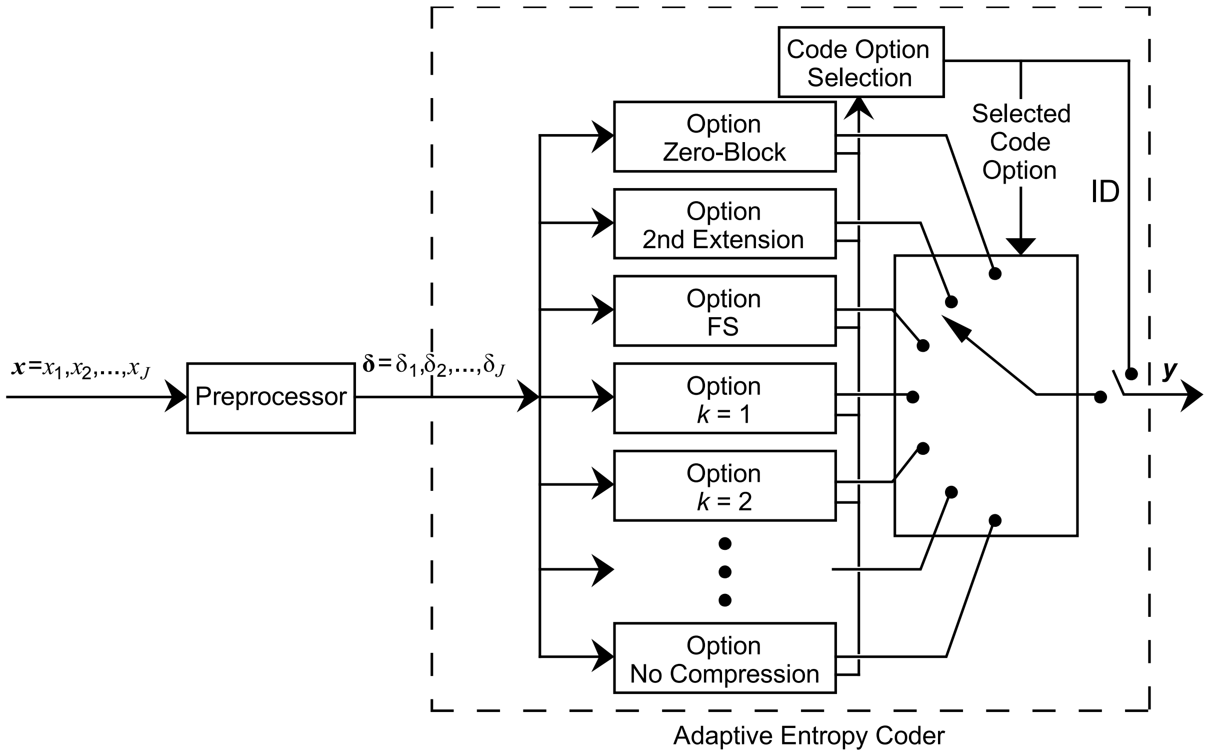


Figure 3-1: The Adaptive Entropy Coder with a Preprocessor

NOTE – Figure 3-1 illustrates the principle of the Adaptive Entropy Coder with a preprocessor; it does not illustrate an implementation.

3.1.2 The basic code selected is a variable-length code that utilizes Rice's adaptive coding technique (refer to reference [C2]). In Rice's coding technique, several algorithms are concurrently applied to a block of J consecutive preprocessed samples, as depicted in figure 3-1. The algorithm option that yields the shortest encoded length for the current block of data is selected for transmission.

3.1.3 The most basic encoding options consist of the Fundamental Sequence (FS) option and Split-Sample options k = 1, 2,…, that also use FS codewords. The various split-sample options enable the adaptation of codeword lengths to the source-data statistics. Two code options, the Second-Extension option and the Zero-Block option, provide more efficient coding than other options when the preprocessed data are highly compressible (low-entropy). There is also a No-Compression option.

3.1.4 The Zero-Block option is a special case in that a single CDS encodes one or more consecutive blocks of J preprocessed samples (see 3.5). In all other single-block options, the CDS produced by the entropy coder encodes a single block of J consecutive preprocessed samples. In all single-block options, the CDS for each block is assembled by encoding each preprocessed sample, but the selection of the best coding option is based on the compressibility of the entire block.

3.1.5 The following variables are required by Rice's adaptive coding technique:

– block size, J (number of samples per block);

– sample resolution, n (number of input bits per sample);

– the ID bit sequence of the selected code option.

3.1.6 The following constraints shall apply to the Entropy Coder's variable-length adaptive coding scheme:

J	=	8, 16, 32, or 64 samples per block;
n	=	Sample resolution with a maximum of 32 bits per sample corresponding to either unsigned digital signal values from 0 to 2^n-1 or to signed values from -2^{n-1} to $2^{n-1}-1$.

3.2 THE FUNDAMENTAL SEQUENCE OPTION

3.2.1 The most basic option is a variable-length FS codeword, which consists of m zeros followed by a one when preprocessed sample $\delta_i = m$. Table 3-1 defines the mapping of preprocessed sample values δ_i to FS codewords.

3.2.2 When the FS option is selected, FS codewords are generated for each preprocessed sample and concatenated to encode the whole input block. Detailed specification of the resulting CDS for a block coded with the FS option is given in 5.2.3.

Table 3-1: Fundamental Sequence Codewords As a Function of the Preprocessed Samples

Preprocessed Sample Values, δ_i	FS Codeword
0	1
1	01
2	001
.	.
.	.
.	.
2^n-1	$\underbrace{0000 \ldots 0000}_{(2^n-1 \text{ zeros})}1$

3.3 SPLIT-SAMPLE OPTIONS

3.3.1 The kth split-sample option is obtained by encoding the $(n-k)$ MSBs from the binary representation of each preprocessed sample δ_i with an FS codeword[1] (see figure 3-2), and then appending the preprocessed sample's k Least Significant Bits (LSBs) uncoded. This produces a varying codeword length.

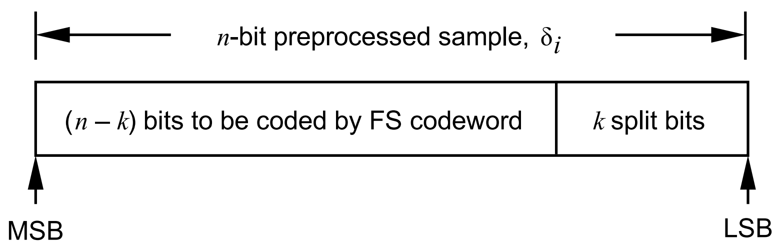


Figure 3-2: Split-Sample Format

3.3.2 The FS option described in 3.2 is a special case of sample splitting where $k = 0$.

3.3.3 Figure 3-2 depicts the encoding of one preprocessed sample, δ_i using the Split-Sample option, but it does not represent the order in which the coded bits for an entire J-sample block are assembled. In the corresponding CDS, the FS codewords for the current block of J preprocessed samples are all transmitted first, followed by the uncoded LSBs for all of the J preprocessed samples. They are preceded by an ID field indicating the value of k. Detailed specification of the resulting CDS for a block coded with the Split-Sample option is given in 5.2.3.

[1] Encoding the $(n-k)$ MSBs with an FS codeword means applying the look-up table in table 3-1 with the numerical value of δ_i replaced by $\lfloor \delta_i / 2^k \rfloor$.

3.4 THE SECOND-EXTENSION OPTION

3.4.1 When the Second-Extension option is selected, each pair of preprocessed samples in a *J*-sample block is transformed and encoded using an FS codeword. A pair of consecutive samples $(\delta_{2j-1}, \delta_{2j})$ from a *J*-sample preprocessed data block are transformed into a single new symbol γ_j by the following equation:

$$\gamma_j = (\delta_{2j-1} + \delta_{2j})(\delta_{2j-1} + \delta_{2j} + 1)/2 + \delta_{2j},$$

where $j = 1, 2, \ldots J/2$. When the first sample in the *J*-sample block is a reference sample, then the γ_j values are calculated using $\delta_1 = 0$.

3.4.2 If the $J/2$ transformed symbols in a block are all smaller than 2^n, the $J/2$ transformed symbols in a block are encoded using FS codewords from table 3-1. If any transformed symbols γ_j are 2^n or higher, the corresponding FS codeword is obtained by extending the mapping in table 3-1 in the obvious manner, that is, the FS codeword consists of γ_j 0s followed by one 1. But it should be noted that the Second-Extension Option is only designed to be a useful option when all of the transformed symbols γ_j are small.

3.4.3 Detailed specification of the resulting CDS for a block coded with the Second-Extension option is given in 5.2.6.

3.5 THE ZERO-BLOCK OPTION

3.5.1 The Zero-Block option is always selected when one or more consecutive blocks of preprocessed samples are all zeros. In this case, a single CDS represents the entire sequence of All-Zeros blocks, unlike other options in which each CDS output from the entropy coder represents only a single block. When a reference sample is required, a block containing a reference sample is considered to be an All-Zeros block if and only if the $(J-1)$ preprocessed symbols $\boldsymbol{\delta} = \delta_2, \ldots \delta_i, \ldots \delta_J$ following the reference sample are all equal to zero.

3.5.2 As described in 4.2.6, there are *r* blocks between consecutive reference samples (or possibly fewer than *r* blocks at the end of the input sequence). Each such sequence of *r* blocks is partitioned into one or more segments of 64 blocks each, except possibly the last, which may be smaller.

3.5.3 Within each segment, each sequence of consecutive All-Zeros blocks is encoded by one FS codeword, as specified in table 3-2. Nonzero blocks that interrupt the sequences of All-Zeros blocks are encoded using one of the single-block options. The encoding of All-Zeros blocks maps the length of the sequence of All-Zeros blocks to a corresponding FS codeword. The Remainder-Of-Segment (ROS) codeword in table 3-2 shall be used to denote that the remainder of a segment consists of five or more All-Zeros blocks. This applies to every segment, including the last segment of each reference sample interval or the last segment at the end of the input sequence, which can be smaller than 64 blocks.

3.5.4 Detailed specification of the resulting CDS for a block coded with the Zero-Block option is given in 5.2.5.

Table 3-2: Zero-Block Fundamental Sequence Codewords As a Function of the Number of Consecutive All-Zeros Blocks

Number of All-Zeros Blocks	FS Codeword
1	1
2	01
3	001
4	0001
ROS	00001
5	000001
6	0000001
7	00000001
8	000000001
.	.
.	.
.	.
63	0000 … 0000000001 (63 0s and a 1)

NOTE – An implementation that does not use the ROS codeword for segments smaller than 64 blocks (end of reference sample interval or end of input sequence) would still produce an encoded bit stream that can be decoded.

3.6 THE NO-COMPRESSION OPTION

3.6.1 The last option is to not apply any data compression. If it is the selected option, the entire preprocessed block of J samples receives an attached identification field but is otherwise unaltered.

3.6.2 Detailed specification of the resulting CDS for a block coded with the Zero-Block option is given in 5.2.4.

3.7 CODE SELECTION

3.7.1 The Adaptive Entropy Coder includes a code selection function, which selects a coding option to minimize the number of encoded bits (including ID bits). The ID bit sequence specifies which option was used to encode the accompanying block of samples. The ID bit sequences are shown in table 5-1.

3.7.2 The Zero-Block option is always selected to encode any sequence of one or more consecutive All-Zeros blocks. This includes the case in which the All-Zeros block contains a reference sample, regardless of whether the reference sample itself is zero.

3.7.3 For blocks that are not All-Zeros, the Adaptive Entropy Coder selects the single-block coding option that minimizes the number of encoded bits (including ID bits) needed to encode the block.

3.7.4 When two or more single-block coding options minimize the length of an encoded block, the option selected for the block should be chosen as follows:

a) the 'no-compression' option should be chosen when it minimizes the encoded length for the block; otherwise,

b) the Second-Extension option should be chosen when it minimizes the encoded length for the block; otherwise,

c) the coding option having the smallest code parameter value k (where the FS option is treated as k=0) should be chosen.

4 PREPROCESSOR

4.1 GENERAL PREPROCESSOR FUNCTION

4.1.1 Two of the factors contributing to the coded bit rate performance (in bits/sample) of this Lossless data compression technique are the amount of correlation removed among data samples in the preprocessing stage, and the coding efficiency of the entropy coder. The function of the preprocessor is to decorrelate data and reformat them into nonnegative integers with the preferred probability distribution. There are situations when a preprocessor is not necessary (see reference [C2]) and may be omitted. Several preprocessing techniques, typically predictive methods, can be used with the Adaptive Entropy Coder.

4.2 PREDICTORS

4.2.1 GENERAL

4.2.2 A predictive preprocessor contains two functions, prediction and mapping, as shown in figure 4-1. The preprocessor subtracts the predicted value, $\hat{x}_i$, from the current data value, x_i. The resultant (n+1)-bit prediction error, Δ_i, is then mapped to an n-bit nonnegative integer value, δ_i, based on the predicted value, $\hat{x}_i$. When a predictor is properly chosen, the prediction error tends to be small, and for some sources, has a probability distribution approaching Laplacian, for which the Adaptive Entropy Coder is optimal. There are several prediction techniques, of which only one, the Unit-Delay Predictor as described in 4.2.5, is presented in this Recommended Standard (see reference [C2] for predictor examples).

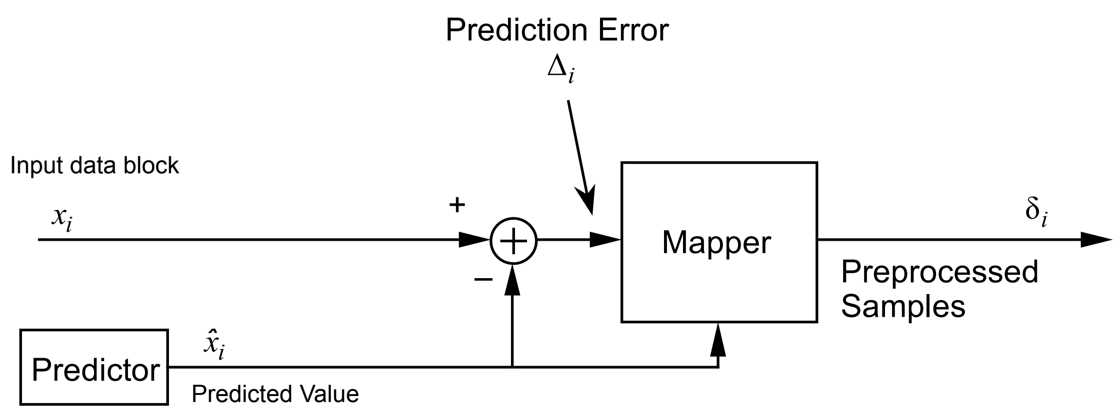


Figure 4-1: A Preprocessor

4.2.3 A ‘bypass’ predictor sets all predicted values to zero, while preserving the mapping stage. This is useful in cases in which prediction is not desired, but the mapper is still useful to map negative samples to positive values that can be encoded by the Adaptive Entropy Coder.

4.2.4 PREDICTION TECHNIQUES

One prediction technique, using the Unit-Delay Predictor, is specified in 4.2.5 below. An application-specific predictor may be used instead of the Unit-Delay Predictor, but such a predictor is unique to the application and is not specified in this Recommended Standard.

4.2.5 THE UNIT-DELAY PREDICTOR

The Unit-Delay Prediction technique uses the one-sample delayed input data signal as the predictor for the current data signal, as illustrated in figure 4-2. That is, the predicted value, $\hat{x}_i$, is equal to the preceding sample value, except for the first sample in a reference interval (as defined in 4.2.6) for which the predicted value is the current sample value, x_i. The prediction error $\Delta_i = x_i - \hat{x}_i$ is passed to the Prediction Error Mapper along with the predicted value $\hat{x}_i$, for mapping to a nonnegative integer δ_i.

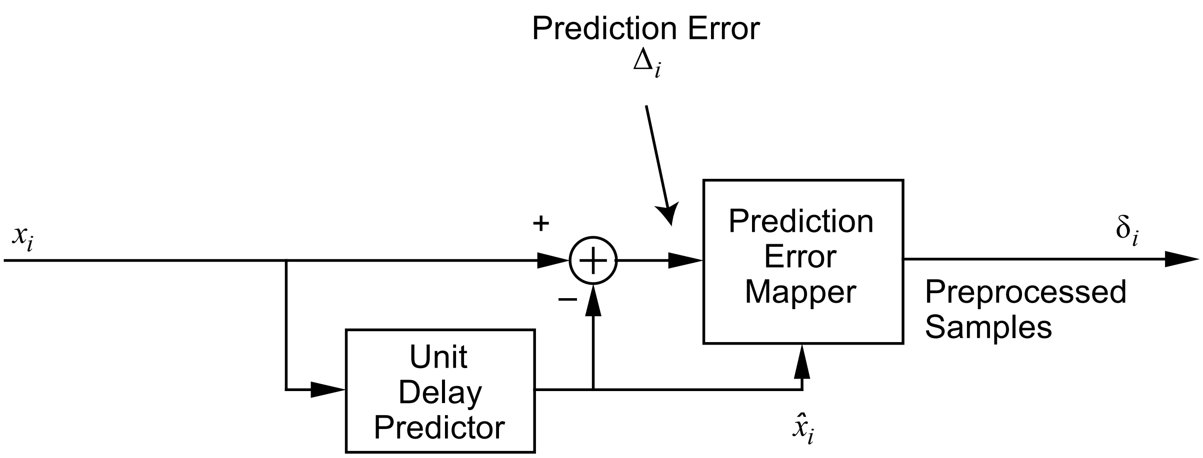


Figure 4-2: Preprocessor Using a Unit-Delay Predictor

4.2.6 REFERENCE SAMPLES FOR PREDICTIVE PREPROCESSORS

A reference sample is an unaltered input data sample upon which succeeding sample prediction is based. When, and only when, a Unit-Delay Predictor or other higher-order predictor that bases its predictions on previous sample values is used, reference samples are required by the decoder in order to invert the preprocessing function. Otherwise, reference samples shall not be employed. Reference samples are always the first sample of a *J*-sample input block, and they pass uncoded directly into the leading position in the corresponding CDS output from the entropy coder, ahead of the same block's (*J*–1) encoded preprocessed samples. The user indicates the frequency of reference sample insertion by specifying the reference sample interval *r* as described in 4.3.

4.3 REFERENCE SAMPLE INTERVAL

The user-specified reference sample interval, r, is limited to a maximum value of 4096 blocks (e.g., 262,144 samples when $J = 64$). When a reference sample is not required by the preprocessor, the parameter r serves to define an interval of input data sample blocks that will be further segmented under the Zero-Block option described in 3.5.

4.4 PREDICTION ERROR MAPPER

The Prediction Error Mapper takes the prediction error values Δ_i and maps them into nonnegative integers δ_i suitable for input to the Adaptive Entropy Coder. With a properly chosen predictor, the most probable value of Δ_i is zero, followed by +1 and –1, +2 and –2, …, etc. The prediction error Δ_i resulting from taking the difference between a sample value, x_i, and a predicted value, $\hat{x}_i$, both n-bit integers, will have an $(n+1)$-bit dynamic range of $[-2^n+1, 2^n-1]$. However, for every predictor value $\hat{x}_i$, there are only 2^n possible prediction error values Δ_i. The smallest prediction error value is the difference between the minimum signal value, x_{min}, and the predictor value, $\hat{x}_i$: $x_{min} - \hat{x}_i$. The largest prediction error value is the difference between the maximum signal value, x_{max}, and the predictor value, $\hat{x}_i$: $x_{max} - \hat{x}_i$. To map the possible 2^n prediction error values into nonnegative integers, the following equation is used:

$$\delta_i = \begin{cases} 2\Delta_i & 0 \le \Delta_i \le \theta_i \\ 2\left|\Delta_i\right| - 1 & -\theta_i \le \Delta_i < 0 \\ \theta_i + \left|\Delta_i\right| & \text{otherwise} \end{cases},$$

where

$$\theta_i \quad = \quad \min(\hat{x}_i - x_{min}, x_{max} - \hat{x}_i).$$

For signed n-bit sample values,

$$x_{min} \quad = \quad -2^{n-1}, \text{ and } x_{max} = 2^{n-1} - 1;$$

for unsigned n-bit sample values,

$$x_{min} \quad = \quad 0, \text{ and } x_{max} = 2^n - 1.$$

5 CODED DATA SET AND PACKET FORMATS

5.1 OVERVIEW

Section 3 specifies how the Adaptive Entropy Coder computes the encoded data within a CDS, but does not specify how the CDS is assembled. An Option Identification (ID) Key is included at the beginning of every CDS to indicate to the decoder the encoding option used for the corresponding data block. Additionally, the detailed formatting of a CDS depends on whether or not the corresponding data block includes a reference sample. CDS Format details are specified in 5.2.

CDSes are packaged into packets, groups of packets, or files, as specified in 5.3, section 6, and section 7, respectively. The requisite packet or file formats allow provision of parameters required in order to transfer the adaptive variable-length losslessly coded data between the coder and the telemetry channel packet formatter, as well as compressor parameters needed for recovering the original data that do not change with every CDS.

5.2 THE CODED DATA SET FORMAT

5.2.1 CODE OPTION IDENTIFICATION KEY

5.2.1.1 Users shall choose to use either the Basic or Restricted set of code options. When sample resolution $n \leq 4$, the use of the Restricted set of code options reduces the number of available coding options, thus allowing the use of shorter ID bit sequences.

5.2.1.2 The ID Field specifies which of the options was used for the accompanying set of samples. The ID-code keys for each of the options are shown in table 5-1.

5.2.1.3 For applications not requiring the full entropy range of performance provided by the specified code options, a subset of the options at the source may be implemented. The ID key in table 5-1 is always required, even if only a subset of the options is used.

Table 5-1: Selected Code Option Identification Key

Code Option	Resolution					
	Basic:	**–**	**–**	**$n \le 8$**	**$8 < n \le 16$**	**$16 < n \le 32$**
	Restricted:	**$n = 1, 2$**	**$n = 3, 4$**	**$4 < n \le 8$**	**$8 < n \le 16$**	**$16 < n \le 32$**
Zero-Block		00	000	0000	00000	000000
Second-Extension		01	001	0001	00001	000001
FS		—	01	001	0001	00001
k=1		—	10	010	0010	00010
k=2		—	—	011	0011	00011
k=3		—	—	100	0100	00100
k=4		—	—	101	0101	00101
k=5		—	—	110	0110	00110
k=6		—	—	—	0111	00111
k=7		—	—	—	1000	01000
k=8		—	—	—	1001	01001
k=9		—	—	—	1010	01010
k=10		—	—	—	1011	01011
k=11		—	—	—	1100	01100
k=12		—	—	—	1101	01101
k=13		—	—	—	1110	01110
k=14		—	—	—	—	01111
k=15		—	—	—	—	10000
⋮		⋮	⋮	⋮	⋮	⋮
k=29		—	—	—	—	11110
No-compression		1	11	111	1111	11111
NOTE – '—' indicates no applicable value						

5.2.2 INCLUSION OF UNCODED REFERENCE SAMPLES IN A CDS

When the preprocessor is present and reference samples are required, the first CDS of the Space Packet Data Field or the first CDS of the compressed file shall contain a reference sample. References shall then be inserted every r blocks as specified in 4.2.6. When the preprocessor is absent, or it does not require a reference sample, the reference sample shall not be inserted in the CDS.

5.2.3 CDS FORMAT FOR FS AND SPLIT-SAMPLE OPTIONS

The CDS format when a Split-Sample option is selected is shown in figure 5-1. Figure 5-1a) shows the case in which there is a reference sample; figure 5-1b) shows the format when no reference sample is present. The CDS has the following structure when a split-sample option is selected: 1) ID bit sequence optionally followed by an n-bit reference sample, 2) compressed data, and 3) concatenated k least-significant bits from each sample. This specification includes the FS option, which is a special case of the Split-Sample option with k=0.

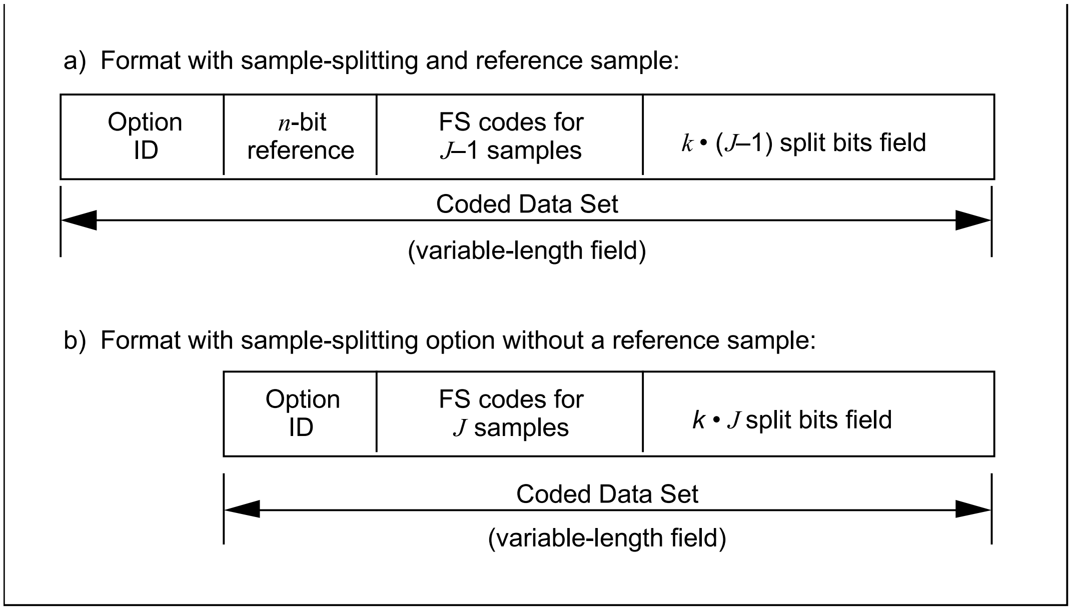


Figure 5-1: CDS Format When a Split-Sample Option Is Selected (Including the Special Case k=0 Equivalent to the FS Option)

5.2.4 CDS FORMAT FOR THE NO-COMPRESSION OPTION

When the no-compression option is selected, the CDS is fixed length, containing the option ID field, optionally followed by an n-bit reference sample, and J or J-1 preprocessed samples. The case in which a reference sample is present is shown in figure 5-2a); the non-reference case is shown in figure 5-2b).

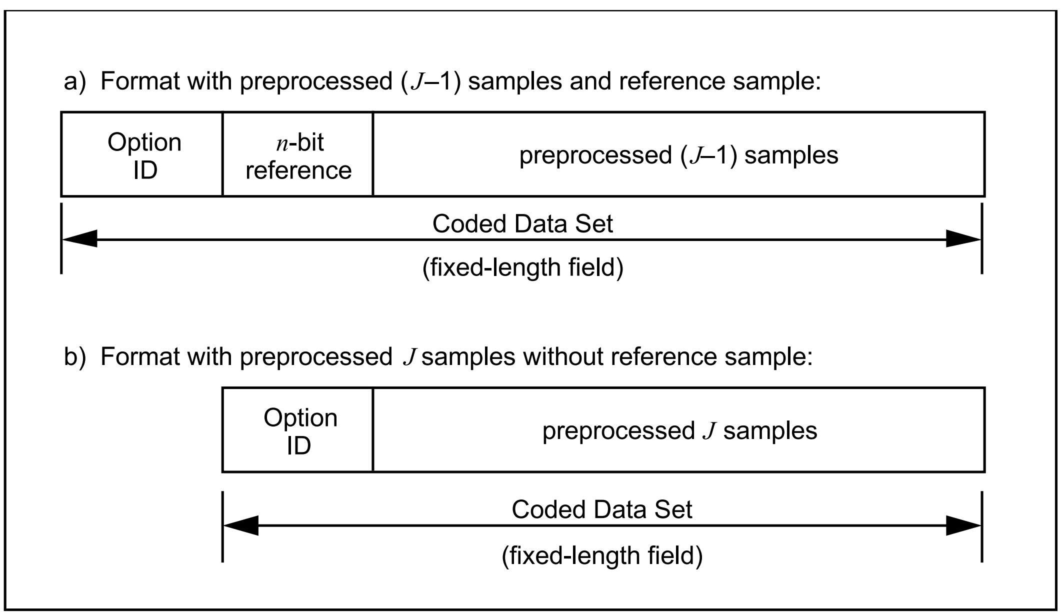


Figure 5-2: CDS Format When the No-Compression Option Is Selected

5.2.5 CDS FORMAT FOR THE ZERO-BLOCK OPTION

When the Zero-Block option is selected, the CDS contains the option ID field, optionally followed by an n-bit reference sample, and a required FS codeword specifying the number of concatenated zero valued blocks or the ROS condition as described in 3.5. The case in which a reference is present is shown in figure 5-3a); the non-reference case is shown in figure 5-3b).

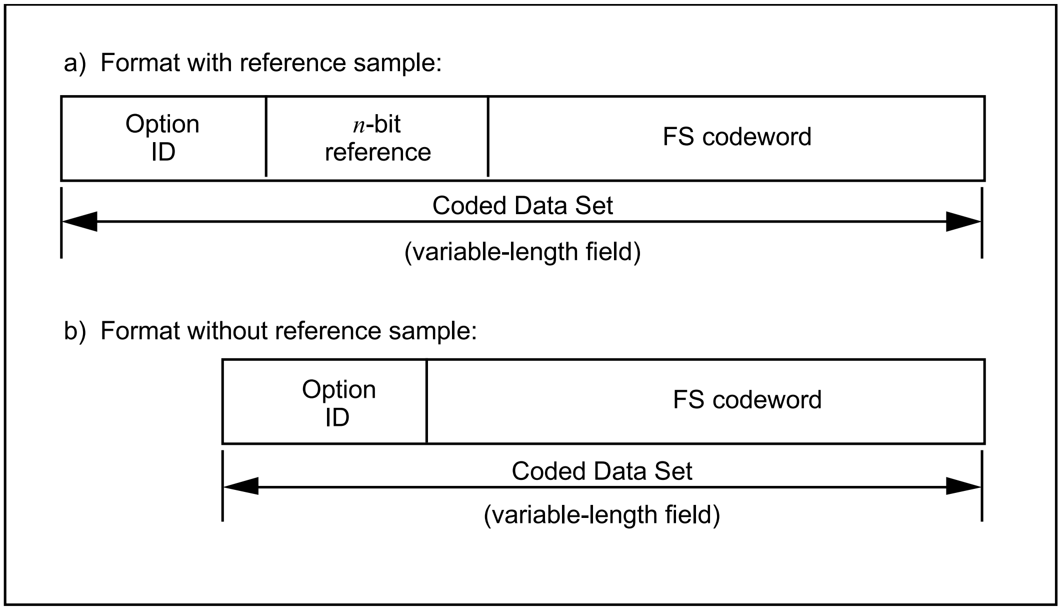


Figure 5-3: CDS Format When the Zero-Block Option Is Selected

5.2.6 CDS FORMAT FOR THE SECOND-EXTENSION OPTION

When the Second-Extension option is selected, the CDS contains the option ID field, optionally followed by an n-bit reference sample, and required FS codewords for $\frac{J}{2}$ transformed pairs of samples. The case in which a reference is present is shown in figure 5-4a); the non-reference case is shown in figure 5-4b). For a block that includes a reference sample, a '0' sample is inserted in front of the J–1 preprocessed samples, so $\frac{J}{2}$ samples are produced after the transformation, and $\frac{J}{2}$ FS codewords are included along with the reference sample itself.

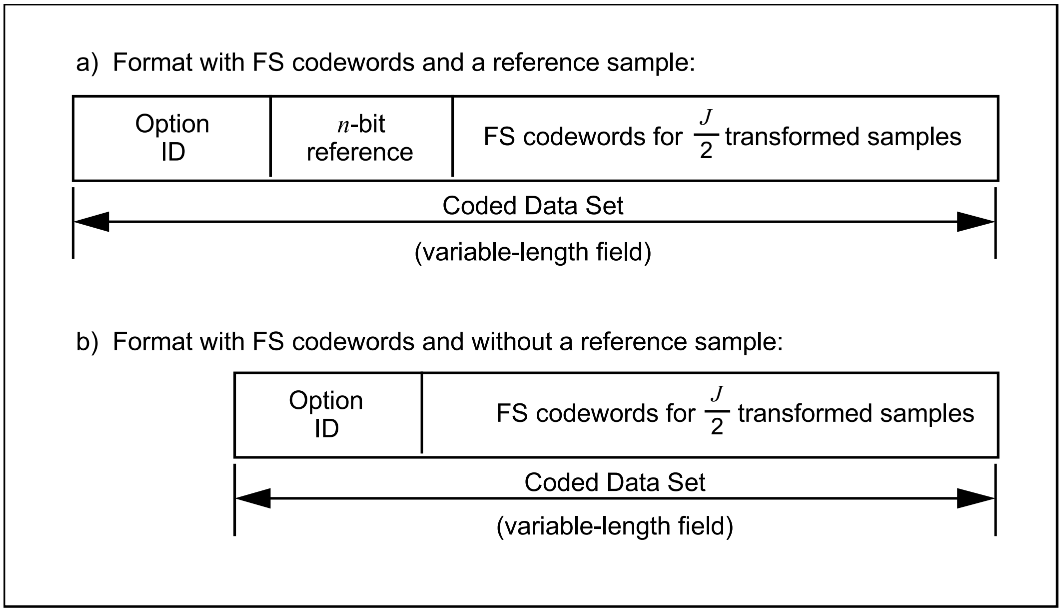


Figure 5-4: CDS Format When the Second-Extension Option Is Selected

5.3 CODED DATA SETS INSERTED INTO SPACE PACKETS

5.3.1 LOSSLESS PACKET FORMAT

When the CCSDS space packet structure (reference [4]) is used to transport the CDSes, the lossless data compression packets shall be formatted as shown in figure 5-5 (see reference [4]). The packet formatter uses the parameter provided by the source data coder to form one or more CDSes to determine the packet size in bytes. Fill bits of zero value may be needed to force the packet to end on a byte boundary.

Packet Primary Header	Secondary Header (optional)	CDS #1 (with n-bit reference)	. . .	CDS #l	Fill Bits

Figure 5-5: Packet Format for *l* CDSes

5.3.2 PACKET REQUIREMENTS

5.3.2.1 A Source Packet Data Field must meet the following requirements:

a) a CDS within a packet must meet the format requirements defined in 5.2;

b) when the reference sample is used, the Source Packet Data Field shall begin with a CDS that contains this reference, followed by one or more additional CDSes; when the reference sample is not required in the preprocessor, or the preprocessor is absent, a reference sample shall not be inserted in the first CDS in the Source Packet Data Field;

c) several CDSes can be put in sequence within a source packet;

d) fill bits are allowed only at the end of the Source Packet Data field, not within the body of compressed data;

e) each packet must end on a byte boundary.

NOTE – Some implementations may require that additional fill bits be added in order to end a packet on an even-numbered byte boundary.

5.3.2.2 Unless the option to use the CIP is chosen (see section 6), in order to decode packets that may include fill bits, several pieces of information must be communicated to the decoder a priori. This information will be mission specific and fixed for a given Application Process Identifier (APID) per mission:

a) l, the number of CDSes that are in a packet;

b) r, the reference sample interval;

c) n, the resolution;

d) J, the number of samples per block;

e) whether the Basic or Restricted set of code options is used (when $n \leq 4$);

f) N, number of samples of the input sequence.

5.3.2.3 A Packet Secondary Header is optional and can be used, for example, to relate observation time and position information to the user (see reference [4]).

5.3.2.4 The use of the Sequence Flags in the Packet Sequence Control Field is optional and can be used, for example, to signal a group of compressed data packets. Their use is governed by reference [4].

6 COMPRESSION IDENTIFICATION PACKET (OPTIONAL)

6.1 COMPRESSION IDENTIFICATION PACKET STRUCTURE

6.1.1 When the compressed data are transmitted as groups of source packets, a Compression Identification Packet (CIP) is an optional packet that, if used, shall precede and provide configuration information for a group of compressed application data packets. The CIP will be transmitted from an application process in space to one or several sink processes on the ground.

6.1.2 The CIP shall be the first packet of the group.

6.1.3 The CIP shall consist of two major fields positioned contiguously in the following sequence: CIP Packet Primary Header and Packet Data Field. (See figure 6-1.)

6.1.4 The CIP shall contain information that would allow the decompressor to be automatically configured to acquire a group of compressed application data packets without the need for managing a priori information. The CIP shall be utilized to configure the decompressor automatically only if there is a reliable system for file transfer.

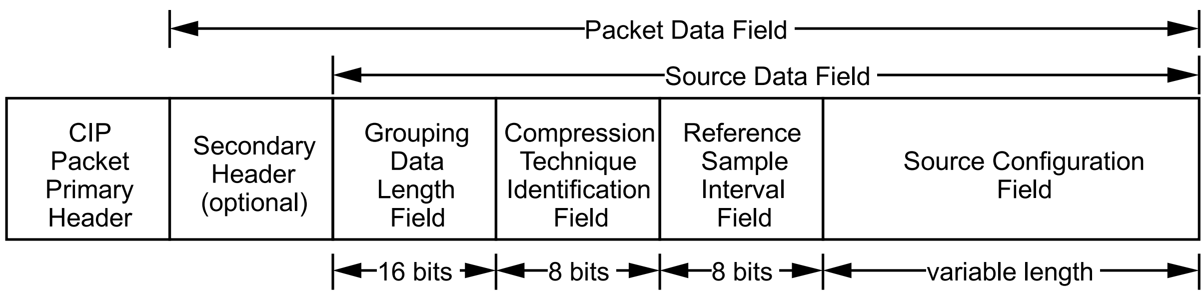


Figure 6-1: CIP Structure

6.2 CIP PACKET PRIMARY HEADER

6.2.1 GENERAL

The CIP Packet Primary Header is mandatory for the CIP and its structure shall conform to the CCSDS Space Packet Protocol Blue Book, reference [4]. The CIP Packet Primary Header Field shall contain the source data APID. The use of the CIP will be mission specific and fixed for a given APID.

6.2.2 SEQUENCE FLAGS

6.2.2.1 The Sequence Flags are in the packet Sequence Control field, as specified in reference [4]. The field is located in the Packet Primary Header of packets encapsulating compressed user data. As indicated below, the field is always '01' for the CIP Primary Header.

6.2.2.2 The Sequence Flags shall be set as follows:

– ‘01’ for the group’s first packet, which is the CIP;

– ‘00’ for the continuing source packets with compressed data of the group;

– ‘10’ for the last source packet with compressed data of the group.

6.2.2.3 For a source packet not belonging to a group of source packets with compressed data, the Sequence Flags shall be set to ‘11’.

6.3 PACKET DATA FIELD

6.3.1 GENERAL

The Packet Data Field of a CIP shall consist of two fields positioned contiguously in the following sequence: Packet Secondary Header (optional) and the Source Data Field.

6.3.2 SECONDARY HEADER (OPTIONAL)

The Secondary Header is a means for placing ancillary data such as time and spacecraft position/attitude information with the CIP.

6.3.3 SOURCE DATA FIELD

6.3.3.1 General

The Source Data Field for the CIP shall consist of four fields positioned contiguously in the following sequence:

	Length (bits)
The Grouping Data Length	16
Compression Technique Identification	8
Reference Sample Interval	8
Source Configuration	(Variable)

6.3.3.2 Grouping Data Length

The Grouping Data Length is a 16-bit field of which the first 4 bits are reserved. The remaining 12 bits of the field shall contain a binary number equal to the number of packets containing compressed data within the group minus one, with the number of packets containing compressed data ranging from 1 to 4096. The number of packets in the group with the CIP included shall range from 2 to 4097.

6.3.3.3 Compression Technique Identification Field

6.3.3.3.1 The Compression Technique Identification (CTI) field shall signal the compression technique in use for the group of source packets identified by the CIP.

6.3.3.3.2 When the no-compression technique for the current group is used, the CTI field shall be set to all zeros.

6.3.3.3.3 Only the Lossless data compression technique is currently defined, and is signaled by the value '00000001' in the CTI field. Other values are reserved for future use by CCSDS and are not permitted.

6.3.3.4 Reference Sample Interval Field

The 8-bit Reference Sample Interval field shall contain a binary number equal to $(r-1)$ mod 256. That is, this field encodes the modulus of $(r-1)$ with respect to divisor 256.

6.3.3.5 Source Configuration Field

6.3.3.5.1 Subfields of the Source Configuration Field

6.3.3.5.1.1 The Source Configuration field shall be partitioned into four subfields, which should appear in the following order: Preprocessor, Entropy Coder, Extended Parameters, and Instrument Configuration Parameters (see figure 6-2). The Preprocessor and Entropy Coder subfields are required, whereas the Instrument Configuration Subfield (ICS) is optional. The Extended Parameters subfield is required whenever any of the following conditions hold:

a) the block length J satisfies J>16;

b) the reference sample interval r satisfies r>256;

c) the Restricted set of code options is used (see 5.2.1.1).

If none of the above conditions holds, then the Extended Parameters subfield shall not be included.

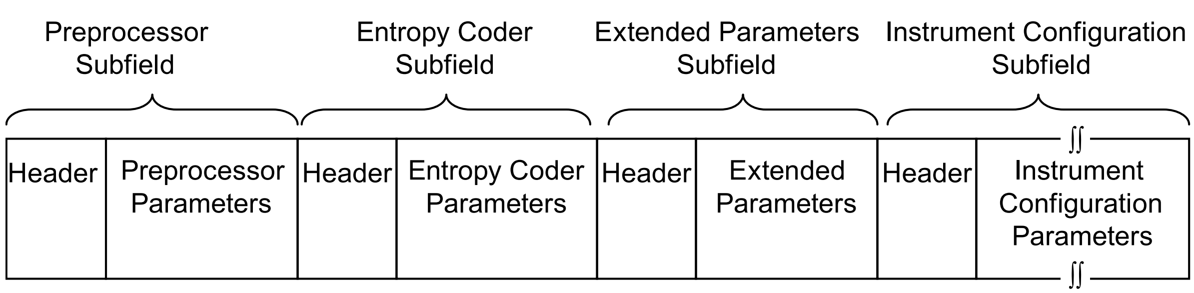


Figure 6-2: Subfields of the Source Configuration Field

6.3.3.5.1.2 Each subfield of the Source Configuration field shall have a header as the first two bits to identify the subfield type. These subfield header bits shall be set as follows:

00 – Preprocessor
01 – Entropy Coder
10 – Instrument Configuration
11 – Extended Parameters

6.3.3.5.2 Preprocessor Subfield

6.3.3.5.2.1 The length of the Preprocessor subfield shall be two bytes, the first two bits of which shall be the header ‘00’ as described in 6.3.3.5.1.2.

6.3.3.5.2.2 The Preprocessor parameters for the Lossless data compressor shall be partitioned into six areas and shall be positioned contiguously following the 2-bit Preprocessor header. (See 3.1 and section 4 for preprocessor parameter definitions.) The six areas are:

a) Preprocessor Status (1 bit)

 0 – absent
 1 – present

 The preprocessor status shall be set to ‘0’ (absent) when the source coder is being used as the block-adaptive encoder defined in 5.4.3.3 of reference [5].

b) Predictor type (3 bits); ignore if preprocessor status is ‘0’:

 000 – bypass predictor or preprocessor absent
 001 – unit delay predictor
 111 – application-specific predictor

 All other codes are reserved by CCSDS for future preprocessing options.

c) Mapper type (2 bits); ignore if preprocessor status is ‘0’:

 00 – Prediction Error mapper described in 4.4 or preprocessor absent
 01 – reserved
 10 – reserved
 11 – application-specific mapper

d) Block size (J) (2 bits):

	Number of Samples/Block
00	J=8
01	J=16
10	J=32 or J=64
11	application specific

NOTE – When J=32 or J=64, the value of J is encoded in the Extended Parameters subfield.

e) Data sense (l bit):

0 – two's complement
1 – positive; mandatory if preprocessor is bypassed or preprocessor absent

f) Input data sample resolution (n) (5 bits):

The 5-bit Input Data Sample field shall contain a binary number equal to the input data sample resolution n minus one, with the data sample resolution ranging from 1 to 32.

6.3.3.5.3 Entropy Coder Subfield

6.3.3.5.3.1 The length of the Entropy Coder subfield shall be two bytes, the first two bits of which shall be the header '01' as described in 6.3.3.5.1.2.

6.3.3.5.3.2 The Entropy Coder parameters subfield shall be partitioned into two areas and shall be positioned contiguously following the 2-bit Entropy Coder header. The two areas are:

a) Data resolution range (2 bits):

00 — Spare
01 — for $n \leq 8$
10 — for $8 < n \leq 16$
11 — for $16 < n \leq 32$

b) Number of CDSes per packet, l (12 bits):

The 12-bit field indicating the number of CDSes per packet (l) shall contain a binary number equal to $l - 1$.

6.3.3.5.4 Instrument Configuration Subfield

The ICS is an instrument-unique field used to address unique instrument configuration parameters. The contents of this field are mission specific and are beyond the scope of this Recommended Standard. If used, the first two bits shall be the header '10' as specified in 6.3.3.5.1.2.

6.3.3.5.5 Extended Parameters

6.3.3.5.5.1 The length of the Extended Parameters subfield shall be two bytes, the first two bits of which shall be the header '11' as described in 6.3.3.5.1.2.

6.3.3.5.5.2 The Extended Parameters subfield shall be partitioned into six areas. The six areas are:

a) Reserved (2 bits):

These two bits shall be set to ‘00’.

b) Block size (J) (4 bits):

	Number of Samples/Block
0000	J=8
0001	J=16
0010	J=32
0011	J=64
1111	application specific

Other values are reserved.

c) Reserved (1bit):

This bit shall be set to ‘0’.

d) Restricted code options flag (see 5.2.1.1) (1 bit):

0 – Basic set of code options are used;

1 – Restricted set of code options are used.

e) Reserved (2 bits):

These two bits shall be set to ‘00’.

f) Reference sample interval extension (4 bits):

This field shall encode the value $\lfloor (r-1)/256 \rfloor$. That is, the largest integer less than or equal to (r–1)/256 shall be encoded.

7 FILE FORMAT

7.1 OVERVIEW

When compressed data are stored or transmitted as a file, the File Format is an optional format that provides information about compression options and defines a structure to store the sequence of CDSes resulting from compressing N input samples.

CCSDS File Delivery Protocol (CFDP) (see reference [6]) is the available CCSDS solution for file transfer over space links.

7.2 FILE STRUCTURE

7.2.1 GENERAL

7.2.1.1 The File Format shall consist on a header specified in 7.2.2, followed by a body specified in 7.2.3 as depicted in figure 7-1.

File Header	File Body

Figure 7-1: File Format

7.2.1.2 The user-selected Output Word Size, measured in bytes, shall be an integer B in the range $1 \leq B \leq 8$.

7.2.2 FILE HEADER

The File Header shall consist of 12 bytes having the structure defined in table 7-1.

Table 7-1: File Header Structure

Field	Width (bits)	Description	Reference
Reserved	1	This field shall have value ‘0’.	
Output Word Size (B)	3	The value *B-1* encoded as a 3-bit unsigned binary integer.	7.2
Preprocessor Status	1	‘0’: Preprocessor absent ‘1’: Preprocessor present	4.1

Field	Width (bits)	Description	Reference
Predictor Type	3	'000': bypass predictor or preprocessor absent '001': unit delay predictor '111': application-specific predictor All other codes are reserved by CCSDS for future preprocessing options.	4.2
Mapper Type	2	'00': Prediction Error mapper or preprocessor absent '01': reserved '10': reserved '11': application-specific mapper	4.4
Data Sense	1	'0': two's complement '1': positive (mandatory if preprocessor is bypassed or preprocessor absent)	4.4
Reserved	8	This field shall have the value '00000000'.	
Input Data Resolution	5	This field shall contain the value *n-1* encoded as a 5-bit unsigned binary integer.	4.4
Reserved	1	This field shall have the value '0'.	
Block Size	2	'00': J=8 '01': J=16 '10': J=32 '11': J=64	3.1
Restricted Code Option	1	'0': Basic set of code options are used; '1': Restricted set of code options are used.	5.2
Reference Sample Interval	12	This field shall contain a binary number equal to r–1, encoded as a 12-bit unsigned binary integer.	4.2.6
Reserved	8	This field shall have the value '00000000'.	
Number of Samples (N)	48	This field shall be set to the total number of compressed input samples that are contained in the file, encoded as the 48-bit unsigned binary integer representation of *N-1*.	

NOTE – The File Header contains information which allows decompression of the CDSes stored in the File Body, without a priori information.

7.2.3 FILE BODY

7.2.3.1 The File Body shall consist of the concatenation of the CDSes (as defined in defined in subsection 5.2.3) resulting from compressing N input samples.

7.2.3.2 Following the last CDS in the compressed file, fill bits shall be appended as needed to reach the next Output Word Size boundary as per 7.2.1.2 so that the compressed file size is a multiple of the Output Word Size (B). Fill bits shall be all 'zeros'.

ANNEX A

SECURITY, SANA, AND PATENT CONSIDERATIONS

(INFORMATIVE)

A1 SECURITY CONSIDERATIONS

A1.1 SECURITY BACKGROUND

It is assumed that security is provided by encryption, authentication methods, and access control to be performed at the application and/or transport layers. Mission and service providers are expected to select from recommended security methods suitable to the specific application profile. Specification of these security methods and other security provisions is outside the scope of this Recommended Standard.

A1.2 SECURITY CONCERNS

Security concerns in the areas of data privacy, integrity, authentication, access control, availability of resources, and auditing are to be addressed in the appropriate layers and are not related to this Recommended Standard. The use of lossless data compression does not affect the proper functioning of methods used to achieve such protection.

The use of lossless data compression slightly improves data integrity because the alteration of even a single bit of compressed data is likely to cause conspicuous and easily detectible corruption of the reconstructed data, thus making it more likely that malicious data alteration will be detected.

A1.3 POTENTIAL THREATS AND ATTACK SCENARIOS

An eavesdropper will not be able to decompress compressed data if proper encryption is performed at a lower layer.

A1.4 CONSEQUENCES OF NOT APPLYING SECURITY

There are no specific security measures prescribed for compressed data. Therefore consequences of not applying security are only imputable to the lack of proper security measures in other layers.

A2 SANA CONSIDERATIONS

The recommendations of this document do not require any action from SANA.

A3 PATENT CONSIDERATIONS

At time of publication, the specifications of this Recommended Standard are not known to be the subject of patent rights. There is currently no known active patent for this standard.[2]

[2] The United States Patent and Trademark Office shows the status of a previously applicable patent (U.S. Patent 5448642) to be 'Expired' at time of publication of the current issue of this Recommended Standard.

ANNEX B

GLOSSARY OF ABBREVIATIONS AND TERMS

(INFORMATIVE)

B1 PURPOSE

This annex defines abbreviations and terms used throughout this Recommended Standard to describe source coding for data compression.

B2 ABBREVIATIONS AND ACRONYMS

APID	application process identifier
CCSDS	Consultative Committee for Space Data Systems
CDS	coded data set
CFDP	CCSDS File Delivery Protocol
CIP	compression identification packet
CTI	compression technique identification
FS	Fundamental Sequence
ICS	instrument configuration subfield
LSB	least significant bit
MSB	most significant bit
ROS	remainder of segment
SANA	Space Assigned Numbers Authority

B3 TERMS

ADAPTIVE ENTROPY CODER: An entropy coder codes the source samples with uniquely decodable codewords that, upon decoding, reconstruct the source samples. With an Adaptive Entropy Coder, the average codeword length also follows closely the information content of the source.

ENTROPY: Entropy is a quantitative measure of the average amount of information per source sample, expressed in bits/sample.

FUNDAMENTAL SEQUENCE: The variable-length FS code represents the nonnegative integer m with a binary codeword of m zeros followed by a 1. Application of the FS code to a block of J samples produces a sequence of J concatenated codewords called the Fundamental Sequence.

RICE'S ADAPTIVE CODING: The basic Rice adaptive coding algorithm chooses the best of several code options to use on a block of data. These options are targeted to be efficient over different ranges of data activity. The options are implemented using a combination of FS coding and the splitting of preprocessed samples into their most-significant and least-significant bit parts.

SAMPLE SPLITTING: Sample splitting is a procedure for separating the binary representation of a sample into two groups of adjacent bits, one for lower-order bits, the other for higher-order bits.

SPLIT BITS: Split bits are the lower-order bits separated by sample splitting from the binary representation of a sample.

ANNEX C

INFORMATIVE REFERENCES

(INFORMATIVE)

[C1] *Organization and Processes for the Consultative Committee for Space Data Systems*. Issue 4. CCSDS Record (Yellow Book), CCSDS A02.1-Y-4. Washington, D.C.: CCSDS, April 2014.

[C2] *Lossless Data Compression*. Issue 3. Report Concerning Space Data System Standards (Green Book), CCSDS 120.0-G-3. Washington, D.C.: CCSDS, April 2013.