Standard ECMA-321
June 2001

ECMA

Standardizing Information and Communication Systems

Streaming Lossless Data Compression Algorithm – (SLDC)

Phone: +41 22 849.60.00 - **Fax:** +41 22 849.60.01 - **URL:** http://www.ecma.ch - **Internet:** helpdesk@ecma.ch

Standard ECMA-321
June 2001

ECMA

Standardizing Information and Communication Systems

Streaming Lossless Data Compression Algorithm – (SLDC)

Phone: +41 22 849.60.00 - **Fax:** +41 22 849.60.01 - **URL:** http://www.ecma.ch - **Internet:** helpdesk@ecma.ch

Brief History

In the past decades, ECMA has published numerous ECMA Standards for magnetic tapes, magnetic tape cassettes and cartridges, as well as for optical disk cartridges. Those media developed recently have a very high physical recording density. In order to make optimal use of the resulting data capacity, lossless compression algorithms have been designed that allow a reduction of the number of bits required for the representation of user data.

These compression algorithms are registered by ECMA, the International Registration Authority established by ISO/IEC. The registration consists in allocating to each registered algorithm a numerical identifier that will be recorded on the medium and, thus, indicate which compression algorithm(s) has been used.

This ECMA Standard is the fourth ECMA Standard for compression algorithms. The three previous standards are:

ECMA-151 Data Compression for Information Interchange – Adaptive Coding with Embedded Dictionary –
ISO/IEC 11558 DCLZ Algorithm (June 1991)

ECMA-159 Data Compression for Information Interchange - Binary Arithmetic Coding Algorithm -
ISO/IEC 12042 (December 1991)

ECMA-222 Adaptive Lossless Data Compression Algorithm – (June 1995)
ISO/IEC 15220

This ECMA Standard ECMA-SLDC is based on ECMA-222. It has been extended by defining a set of control symbols, that identify:

- record boundaries in user data
- locations in the Encoded Data Stream at which the history buffer is reset
- locations in the Encoded Data Stream at which pad bits are inserted up to the next 32-bit boundary
- sections in the Encoded Data Stream that contain compressed or uncompressed data

This compression algorithm allows for records of different size and compressibility, along with File Marks, to be efficiently encoded into an output stream in which little or no additional control information is needed to later decode user data.

All ECMA Standards listed above have been adopted by ISO/IEC as International Standards. This ECMA Standard will also be contributed to ISO/IEC for adoption as an International Standard under the fast-track procedure.

This ECMA Standard has been adopted by the ECMA General Assembly of June 2001.

Table of contents

1 Scope

This ECMA Standard specifies a lossless compression algorithm to reduce the number of 8-bit bytes required to represent data records and File Marks. The algorithm is known as Streaming Lossless Data Compression algorithm (SLDC).

One buffer size (1 024 bytes) is specified.

The numerical identifier according to ISO/IEC 11576 allocated to this algorithm is 6.

2 Conformance

A compression algorithm shall be in conformance with this ECMA Standard if its Encoded Data Stream satisfies the requirements of this ECMA Standard.

3 Reference

ISO/IEC 11576:1993, Information Technology - Procedure for the Registration of Algorithms for the Lossless Compression of Data.

4 Definitions

4.1 Access Point

A location in the Encoded Data Stream at which data may be decoded.

4.2 Control Symbol

A Control Symbol may change the compression scheme, reset the History Buffer, mark the end of a Record, indicate a File Mark, or indicate the termination of an Encoded Data Stream.

4.3 Copy Pointer

A part of the Encoded Data Stream output in scheme 1 that replaces a string of data bytes with a specification of a Matching String.

4.4 data byte

An element of user data that is to be encoded.

4.5 Data Symbol

An element of an Encoded Record that represents one or more data bytes.

4.6 Displacement Field

A field in the Copy Pointer that specifies the location within the History Buffer of the first byte of a Matching String.

4.7 Encoded Data Stream

The output stream after encoding User Data.

4.8 Encoded Record

The output stream after encoding one Record of user data.

4.9 End Marker

A Control Symbol that denotes termination of an Encoded Data Stream.

4.10 End Of Record Symbol (EOR Symbol)

A Control Symbol that denotes the end of a Record in the Encoded Data Stream.

4.11 File Mark

A recorded element used to mark organisational boundaries (e.g. directory boundaries) in user data.

4.12 File Mark Symbol

A Control Symbol in Encoded Data Stream that denotes a File Mark in user data.

4.13 Flush Symbol

A Control Symbol that, if required, is followed by Pad to make the size of the Encoded Data Stream an integer multiple of 32 bits.

4.14 History Buffer

A data structure where incoming data bytes are stored for use by scheme 1 compression and decompression.

4.15 Literal 1

A part of the Encoded Data Stream, output in scheme 1, that represents a single data byte not encoded into any Copy Pointer.

4.16 Literal 2

A part of the Encoded Data Stream, output in scheme 2, that represents a single data byte.

4.17 Matching String

A sequence of two or more bytes in the History Buffer that is identical with a sequence of bytes in the user data.

4.18 Match Count

The length, in bytes, of a Matching String.

4.19 Match Count Field

That part of a Copy Pointer that specifies the Match Count.

4.20 Pad

A number of bits inserted into the Encoded Data Stream so that the size of Encoded Data Stream is an integer multiple of 32 bits.

4.21 Record

An element of user data that contains at least one data byte.

4.22 Record Segment

A section of a Record encoded in a given scheme.

4.23 Reset X Symbol

A generic reference to either the Reset 1 Symbol or the Reset 2 Symbol.

4.24 Reset 1 Symbol

A Control Symbol that indicates History Buffer reset, and that subsequent symbols are encoded in scheme 1.

4.25 Reset 2 Symbol

A Control Symbol that indicates History Buffer reset, and that subsequent symbols are encoded in scheme 2.

4.26 scheme 1

A compression scheme that uses a History Buffer to achieve data compression.

4.27 Scheme 1 Symbol

A Control Symbol that indicates subsequent Data Symbols are either Copy Pointers or Literal 1s.

4.28 scheme 2

A packing scheme designed to encode uncompressible data with minimal expansion.

4.29 Scheme 2 Symbol

A Control Symbol that indicates subsequent Data Symbols are encoded in scheme 2.

4.30 user data

Information that is to be encoded, according to this compression algorithm.

5 Conventions and Notation

5.1 Representation of numbers

The following conventions and notations apply in this document unless otherwise stated.

- The setting of bits is denoted by ZERO or ONE.
- Numbers in binary notation and bit combinations are strings of digits represented by ZEROs and ONEs with the most significant bit to the left.
- Letters and digits in parentheses represent numbers in hexadecimal notation.
- All other numbers are in decimal form

5.2 Names

The names of basic elements, e.g. specific fields, are written with a capital initial letter.

6 Acronyms

EOR	End Of Record
lsb	least significant bit
msb	most significant bit

7 Algorithm Overview

User data that is to be compressed according to this ECMA Standard consists of Records and File Marks. Records consist of 8-bit data bytes, and may be of any non-zero length.

Data bytes may be encoded in either scheme 1 or scheme 2.

7.1 Scheme 1 Encoding

There may exist within Records repeating strings of two or more data bytes such that information about the length and position of one string may be substituted in place of a subsequent copy or copies of that same string. This information is known as a Copy Pointer. This ECMA Standard allows Copy Pointer substitution when corresponding bytes of the two strings are offset by 1 to 1 023 data bytes within user data. Where string matches occur, data compression is possible, and the number of bits of encoded data can be less than the number of bits of user data, and data compression is possible. Any data bytes that are part of a repeated string may be encoded as a Copy Pointer. Any data byte that is not encoded as a Copy Pointer is encoded as a Literal 1, in which a leading bit set to ZERO is added to the data byte, thereby indicating that this is a Literal 1. Regions over which Copy Pointers and literal values are encoded are defined as being encoded according to scheme 1. Scheme 1 encoding is identical with that of ECMA-222, except for the addition of Control Symbols. These are both implementations of the Lempel-Ziv 1 (LZ1) class of data compression algorithms. Following a Reset 1 Symbol or a Scheme 1 Symbol, all bytes of user data shall be encoded according to scheme 1.

7.2 Scheme 2 Encoding

There may also exist within user data, regions in which few such repeating strings exist. Where there are no repeating strings, scheme1 encoding requires a 9-bit Literal 1value in the Encoded Data Stream for every data byte. This results in an Encoded Data Stream that has 12,5 % more bits than the user data. In order to avoid this data expansion, scheme 2 encoding may be used. In scheme 2 encoding, data bytes are copied to the output bit stream. In order for a decoder to distinguish a data byte set to (FF) from a Control Symbol, a trailing bit set to ZERO is encoded following every data byte of (FF). For random data, this tends to produce an Encoded Data Stream that has about 0,05 % more bits than the user data. Following a Reset 2 Symbol or a Scheme 2 Symbol, all bytes of user data shall be encoded according to scheme 2.

7.3 History Buffer

Matching strings are found within a 1 024-byte History Buffer. Prior to a Reset X Symbol in the Encoded Data Stream, the History Buffer is undefined. Immediately following a Reset X Symbol, the History Buffer is defined as containing no data.

As the first 1 024 data bytes following a Reset X Symbol are recorded, each byte is stored in a subsequent location in the History Buffer, from 0 to 1 023. For each data byte N, comparisons may be made with each of the data bytes at locations 0 to N-1 to test for Matching Strings.

Once the History Buffer is filled, new bytes replace previously stored bytes in locations 0 to 1 023. The storage location wraps from 1 023 to 0. For a data byte stored at location N, comparisons may be made with each of the data bytes at locations other than N, to test for Matching Strings. Matching Strings may wrap around the end of the History Buffer (e.g. Offset 1 022, Length 10).

By updating the History Buffer identically during decoding, the decoder History Buffer shall be identical, after outputting any specific data byte, with the encoder History Buffer after encoding that same data byte. It is, therefore, not necessary to separately include history content information within the Encoded Data Stream.

This ECMA Standard does not specify the conditions under which to reset the History Buffer, switch between scheme 1 and scheme 2, or flush to a 32-bit boundary.

8 Encoding Specification

8.1 User Data

User data shall consist of Records and File Marks. Records may contain any number of 8-bit data bytes. File Marks may be used to mark organisational boundaries (e.g. directory boundaries) in user data.

8.2 History Buffer

A History Buffer shall be filled with 1 024 byte locations, numbered 0 to 1 023. For any data byte being recorded, the History Buffer may contain the 1 023 preceding data bytes. The content of the History Buffer is initially undefined.

Immediately following a Reset X Symbol, the History Buffer is said to contain no data. As each byte N of the first 1 024 bytes are recorded, that byte is stored to location N, from 0 to 1 023, and History Buffer locations 0 to N-1 may be examined to find matching data bytes.

After a data byte is recorded in location 1 023, the next data byte is recorded in location 0, replacing the first data byte that was stored there. As each subsequent data byte is stored to location N, all History Buffer locations other than location N may be examined to find matching data bytes.

Data bytes shall be recorded sequentially into the History Buffer regardless of Record boundaries, File Marks, or encoding scheme. A Reset X Symbol causes the next data byte to be stored to location 0, regardless of the location to which any previous byte was stored.

8.3 Encoded Data Stream

An Encoded Data Stream is a stream of Data Symbols, Control Symbols, and Pads. These shall be packed into consecutive bytes, starting with the most significant bit of byte 0. Figure 1 shows a Control Symbol followed by Data Symbol packed into bytes.

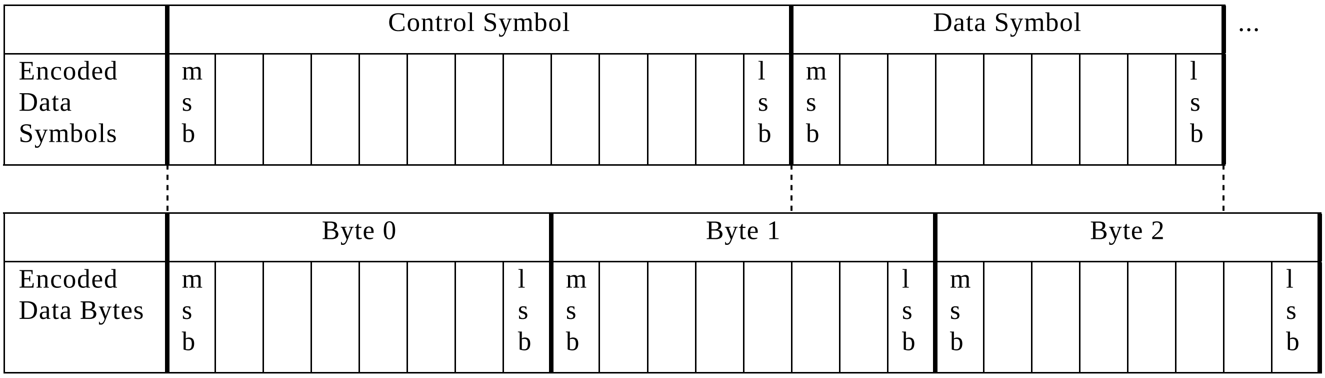

Figure 1 – Encoded Data Stream, packing into bytes

NOTE

Although an Encoded Data Stream is filled with Pad to 32-bit boundaries, packing is expressed by 8-bit bytes. This allows byte ordering within larger words to be specified by any application that uses this compression algorithm.

8.3.1 Access Point

An Access Point is a location in the Encoded Data Stream at which data may be decoded, starting with either a File Mark or the first data byte of a Record. An Access Point must meet the following three requirements.

- It shall occur at a 32-bit boundary. (Previous data shall be padded).
- A Reset X Symbol shall precede the first Data Symbol following the Access Point.
- The first Data Symbol following the Access Point shall represent the first data byte in a Record.

8.4 Data Symbols

Data Symbols shall be used to represent bytes of user data. The Data Symbols are Literal 1, Copy Pointer, or Literal 2.

Following either a Scheme 1 Symbol or a Reset 1 Symbol, only Literal 1 and Copy Pointer Symbols shall be used to represent data bytes (scheme 1 encoding).

Following either a Scheme 2 Symbol or a Reset 2 Symbol, only Literal 2 Symbols shall be used to represent data bytes (scheme 2 encoding).

8.4.1 Literal 1 Data Symbols

Literal 1 Data Symbols shall be used in scheme 1 encoding to represent all data bytes that are not encoded as part of a Copy Pointer. They shall be 9-bit Symbols, consisting of a leading bit set to ZERO and a copy of the data byte, as shown in figure 2.

8	7	6	5	4	3	2	1	0
ZERO	data byte							

Figure 2 – Literal 1 Data Symbol

8.4.2 Copy Pointer Data Symbols

A Copy Pointer Data Symbol may be used in scheme 1 encoding to represent a string of data bytes for which a Matching String exists in the History Buffer. A Copy Pointer shall consist of a leading bit set to ONE, an M-bit Match Count Field, and a 10-bit Displacement Field, as shown in figure 3.

NOTE

This ECMA Standard does not require that all Matching Strings be represented by Copy Pointer Data Symbols, nor does it specify a priority when multiple Matching Strings occur.

M+10	M+9 to 10	9	8	7	6	5	4	3	2	1	0
ONE	Match Count Field	Displacement Field									

Figure 3 – Copy Pointer Data Symbol

The Displacement Field shall contain the location in the History Buffer of the first data byte of the Matching String.

The Match Count Field shall have the length and value specified in Table 1, that indicates the length of the Matching String.

Table 1 – Match Count Field values

Match String Length (bytes)	Match Count Field value
2	0 0
3	0 1
4	10 00
5	10 01
6	10 10
7	10 11
8	110 000
9	110 001
:	:
15	110 111
16	1110 0000
17	1110 0001
:	:
31	1110 1111
32	1111 00000000
33	1111 00000001
:	:
270	1111 11101110
271	1111 11101111
Reserved and Control Symbols	1111 11110000 : 1111 11111111

A Copy Pointer may only replace data bytes from a single Record so that an EOR Symbol in the Encoded Data Stream precisely indicates the end of one Record.

A Copy Pointer may refer to a Matching String that includes data bytes from more than one Record, that span File Marks, or that were encoded in scheme 2.

The second byte of the Matching String may not exist in the History Buffer when the match begins (e.g. the first byte of the string to be replaced may become the second byte of a matching string). Conversely, the first bytes of a Matching String may be overwritten by the bytes of a string being replaced by a Copy Pointer as they are entered sequentially into the History Buffer. Thus, the Matching String and the string being replaced need not coexist in their entirety in the History Buffer at the same time.

8.4.3 Literal 2 Data Symbols

Literal 2 Data Symbols shall be used in scheme 2 encoding to represent all data bytes. For data bytes set to (00) to (FE), they shall be 8-bit Symbols, consisting only of a copy of the data byte, as shown in figure 4. For data bytes set to (FF), they shall be 9-bit Symbols, consisting of a copy of the data byte followed by a ZERO, as shown in figure 5.

7	6	5	4	3	2	1	0
data byte							

Figure 4 – Literal 1 Data Symbol, data byte (00) to (FE)

8	7	6	5	4	3	2	1	0
data byte								ZERO

Figure 5 – Literal 1 Data Symbol, data byte (FF)

NOTE

The added ZERO following the data byte (FF) allows a decoder to distinguish between a Literal 2 and a Control Symbol, since all Control Symbols have nine leading bits set to ONE.

8.5 Control Symbols

Control Symbols are inserted into the Encoded Data Stream so that a decoder can correctly decode the Data Symbols. Table 1 lists the names and values of the Control Symbols. The column labelled Pad indicates whether this Symbol is followed by a Pad set to all ZEROs or all ONES. If the column labelled Pad has a value of None, then there is no Pad, and subsequent Symbols are packed into bytes normally.

Table 1 – Control Symbol Values

Control Symbol	Value	Pad
Flush	1 1111 1111 0000	ZEROs
Scheme 1	1 1111 1111 0001	None
Scheme 2	1 1111 1111 0010	None
File Mark	1 1111 1111 0011	ZEROs
EOR	1 1111 1111 0100	ZEROs
Reset 1	1 1111 1111 0101	None
Reset 2	1 1111 1111 0110	None
End Marker	1 1111 1111 1111	ONEs

A Flush Symbol may be used to make the size of the Encoded Data Stream an integer multiple of 32 bits.

A Scheme 1 Symbol indicates that following Data Symbols are encoded in scheme 1.

A Scheme 2 Symbol indicates that following Data Symbols are encoded in scheme 2.

A File Mark Symbol represents a File Mark in user data.

An EOR Symbol shall follow the Data Symbol that encodes one or more bytes up to and including the last data byte of a Record, and precede the first Data Symbol of the next Record.

A Reset 1 Symbol indicates that the following Data Symbols are encoded in scheme 1, and that the History Buffer is reset.

A Reset 2 Symbol indicates that the following Data Symbols are encoded in scheme 2, and that the History Buffer is reset.

An End Marker shall be used to mark the end of an Encoded Data Stream, and shall occur only outside of any Encoded Records.

The following requirements apply to sequences of Control Symbols and Data Symbols:

- A Flush Control Symbol, followed by Pad, may be inserted into an Encoded Data Stream at any location.
- File Mark and End Marker Symbols shall begin on 32-bit boundaries, and may not be located within Encoded Records.
- An empty set of user data shall be indicated by encoding an End Marker Symbol prior to any other Symbols.
- Following an End Marker Symbol, the History Buffer shall be undefined at the beginning of another Encoded Data Stream.
- A Reset X or Schem X Symbol that occurs outside an Encoded Record shall be followed immediately by either a Flush Symbol or a Data Symbol.

8.6 Pad

A Pad may be from 0 to 31 bits in length. The bits of a Pad shall be set to either all ZEROs or all ONEs, as specified in Table 1.

NOTE

A Flush Symbol that starts fewer than 13 bits before a 32-bit boundary will require Pad up to the next 32-bit boundary. The entire Flush and Pad can therefore be from 13 to 44 bits.

Free printed copies can be ordered from:
ECMA
114 Rue du Rhône
CH-1204 Geneva
Switzerland

Fax: +41 22 849.60.01
Email: documents@ecma.ch

Files of this Standard can be freely downloaded from the ECMA web site (www.ecma.ch). This site gives full information on ECMA, ECMA activities, ECMA Standards and Technical Reports.

ECMA
114 Rue du Rhône
CH-1204 Geneva
Switzerland

See inside cover page for obtaining further soft or hard copies.