

INTERNATIONAL STANDARD

ISO/IEC
15200

First edition
1996-12-01

Information technology — Adaptive Lossless Data Compression algorithm (ALDC)

*Technologies de l'information — Algorithme de compression de données
d'adaptation sans pertes (ALDC)*



Reference number
ISO/IEC 15200:1996(E)
Documents
AVA Document Sharing Hub

Contents

Foreword	iii
Introduction	iv
1 Scope	1
2 Conformance	1
3 Normative Reference	1
4 Definitions	1
4.1 Compressed Data Stream	1
4.2 Copy Pointer	1
4.3 Current Address	1
4.4 Data Byte.	1
4.5 Displacement Field.	1
4.6 End Marker	1
4.7 History Buffer.	1
4.8 Literal.	1
4.9 Matching String	1
4.10 Match Count	1
4.11 Match Count Field.	1
4.12 Pad Bits	1
5 Conventions and Notations	1
5.1 Representation of numbers	1
5.2 Names	2
6 ALDC compression algorithm	2
6.1 Encoding description for a 512-byte History Buffer	2
6.2 Description of the Compressed Data Stream	3
Annexes	
A - ALDC encoding format	5
B - ALDC Overview	7
C - ALDC Encoding Flow Chart	8
D - Bibliography	12

© ISO/IEC 1996

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland
Printed in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75% of the national bodies casting a vote.

International Standard ISO/IEC DIS 15200 was prepared by ECMA (as ECMA-222) and was adopted, under a special “fast-track procedure”, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, in parallel with its approval by national bodies of ISO and IEC.

Annexes A to D are for information only.

Introduction

In the past decades ISO/IEC have published numerous International 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 which 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 which will be recorded on the medium and, thus, indicate which compression algorithm(s) has been used.

This International Standard is the third one for lossless compression algorithms. The two previous International Standards are:

- | | |
|--------------------|---|
| ISO/IEC 11558:1992 | <i>Information technology - Data compression for information interchange - Adaptive coding with embedded dictionary - DCLZ algorithm.</i> |
| ISO/IEC 12042:1993 | <i>Information technology - Data compression for information interchange - Binary arithmetic coding algorithm.</i> |

Information technology - Adaptive Lossless Data Compression algorithm (ALDC)

1 Scope

This International Standard specifies a lossless compression algorithm to reduce the number of bytes required to represent data. The algorithm is known as Adaptive Lossless Data Compression algorithm (ALDC). The numerical identifiers according to ISO/IEC 11576 allocated to this algorithm are:

ALDC 512-Byte History Buffer: 3

ALDC 1024-Byte History Buffer: 4

ALDC 2048-Byte History Buffer: 5

2 Conformance

A compression algorithm shall be in conformance with this International Standard if its output data stream satisfies the requirements of this International Standard.

3 Normative Reference

The following standard contains provisions which, through reference in this text, constitute provisions of this International Standard. At the time of publication, the edition indicated was valid. All standards are subject to revision, and parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent edition of the standard indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 11576:1993, *Information technology - Procedure for the registration of algorithms for the lossless compression of data*.

4 Definitions

For the purposes of this International Standard, the following definitions apply.

- 4.1 Compressed Data Stream:** The output stream after encoding.
- 4.2 Copy Pointer:** A part of the Compressed Data Stream which represents a group of two or more consecutive bytes for which there already exists an identical group in the History Buffer. It comprises a Length Code Field and a Displacement Field.
- 4.3 Current Address:** The location within the History Buffer where the Data Byte is written.
- 4.4 Data Byte:** The current byte of incoming data which is written into the History Buffer and is compared to all data bytes previously written into the History Buffer.
- 4.5 Displacement Field:** That part of the Copy Pointer which specifies the location within the History Buffer of the first byte of a Matching String.
- 4.6 End Marker:** A string of 12 ONES indicating the end of the Compressed Data Stream.
- 4.7 History Buffer:** A data structure where incoming data bytes are stored for use in the compression and decompression process.
- 4.8 Literal:** A Data Byte for which no match was found in the History Buffer.
- 4.9 Matching String:** A sequence of bytes in the incoming data which is identical with a sequence of bytes in the History Buffer.
- 4.10 Match Count:** The number of bytes in a Matching String.
- 4.11 Match Count Field:** That part of the Copy Pointer which specifies the number of consecutive bytes for which a match was found in the History Buffer.
- 4.12 Pad Bits:** Bits set to ZERO and included in the Compressed Data Stream, as required, to maintain an 8-bit byte boundary.

5 Conventions and Notations

5.1 Representation of numbers

The following conventions and notations apply in this International Standard, unless otherwise stated.

- The setting of binary bits is denoted by ZERO and ONE.
- Numbers in binary notation and bit combinations are represented by ZEROs and ONEs with the most significant bit to the left.
- All other numbers shall be in decimal form.

5.2 Names

The names of entities are given with a capital initial letter.

6 ALDC compression algorithm

An overview of the ALDC encoding process is given in annex B, and a flow chart is contained in annex C.

6.1 Encoding description for a 512-byte History Buffer

At the start of encoding, all bytes of the History Buffer shall be reset to all ZEROs. Data bytes shall be stored in sequence in the History Buffer, starting with a Current Address of 0.

The encoder processes the incoming data stream one byte at a time. The current byte being processed is referred to as the Data Byte. When a Data Byte is received from the input data stream, it shall be written into the History Buffer at the Current Address. Then the Current Address shall be incremented by 1. If it exceeds the maximum address, which is 511 for a History Buffer size of 512 bytes, it shall be reset to 0.

Step 1

The Data Byte shall be compared with each byte previously written into the History Buffer to identify any identical bytes.

Step 2

If the Data Byte does not match any byte in the History Buffer, the process shall continue at Step 6.

- If the Data Byte matches one or more bytes in the History Buffer, for every matching byte it shall be noted whether this matching byte is a continuation of a previous sequence of matching bytes or not.
- If it is not a continuation, the Displacement Field of the matching byte shall be noted and recorded as having a Match Count of one byte.
- If the matching byte is a continuation of a previous string, the Match Count for that string shall be incremented by 1.

Step 3

If a Match Count equals 271, the corresponding bytes shall be identified by a Copy Pointer, which shall be added to the Compressed Data Stream. Its Match Count Field and Displacement Field shall be specified as defined in 6.2. The next Data Byte shall then be read and the process shall continue at Step 1.

If there is no more data to be read, the process shall continue at Step 7.

Note: The value of 271 was chosen for implementation reasons.

Step 4

If the Match Count has not reached 271, any pending Matching Strings shall be checked to see if any are continued by the Data Byte.

- If none of the previous Matching Strings is continued and if any of the previous Matching Strings consists of two or more bytes, that Matching String having the lowest Displacement Field shall be identified by a Copy Pointer, which shall be added to the Compressed Data Stream. The next Data Byte shall then be read. If there is no more data to be read, the process shall continue at Step 7.
- If no previous Matching Strings are continued and the previous matches were only 1-byte matches, the previous 1-byte match shall be identified as a Literal and shall be added to the Compressed Data Stream as defined in 6.2. The next Data Byte shall then be read. If there is no more data to be read, the process shall continue at Step 7.

Step 5

If there are no Matching Strings with a Match Count of 271 and there is the continuation of at least 1 previous Matching String, the next Data Byte shall be read. The process shall continue at Step 1.