



ISO/IEC JTC1/SC7
Software Engineering
Secretariat: CANADA (SCC)

ISO/IEC JTC1/SC7 N1544
July 14, 1996

DOC TYPE: CD Ballot

TITLE: CD 15475-3: Information Technology - Software Engineering Data Definition and Interchange - Transfer Format - Part 3: Encoding ENCODING.1

SOURCE: WG11

PROJECT: 07.28.2.3

STATUS: First CD ballot

REFERENCES: Resolution 404, SC7 N1313, N1539

ACTION ID: ACT

DUE DATE: 1996-10-28. It would be most appreciated that a copy of the comments, if any, be send to the project 07.28 editor, Mr Jean Bérubé (jberube@ibm.net) before 1996-10-18.

DISTRIBUTION: P, O and L

MEDIUM: E-Mail - uuencoded compressed Word ; Diskette - compressed Word

NO. OF PAGES: 75 [202 kB]

DISK NUMBER: 8



| | |
|------------------------------------|--|
| ISO/IEC JTC1/SC7 CD 15475-3 | |
| Date 1996-07-14 | Reference number ISO/JTC 1/SC 7 N 1544 |
| Supersedes document SC 7 N 1313 | |

THIS DOCUMENT IS STILL UNDER STUDY AND SUBJECT TO CHANGE. IT SHOULD NOT BE USED FOR REFERENCE PURPOSES.

| | |
|---|--|
| ISO/JTC 1/SC 7 Committee Title Software Engineering Secretariat: Standards Council of Canada (SCC) | Circulated to P- and O-members, and to technical committees and organizations in liaison for: X discussion at Ottawa on 1996-10-21 X comment by 1996-10-21 X voting by (P-members only) 1996-10-28 Please return all votes and comments in electronic form directly to the SC 7 Secretariat by the due date indicated. |
|---|--|

ISO/IEC JTC1/SC7

Title: Information Technology - Software Engineering Data Definition and Interchange - Transfer Format
Part 3: Encoding ENCODING.1

Project: 07.28.2.3

Introductory note: See Attachements

Medium: E-Mail - uuencoded compressed Word ; Diskette - compressed Word

No. of pages: 65



| | |
|--|--|
| Vote on Committee Draft ISO/IEC 15475-3 | |
| Date of circulation 1996-07-14 | Reference number ISO/JTC 1/SC 7 N 1544 |
| Closing date 1996-10-28 | |

| | |
|---|--|
| ISO/JTC 1/SC 7 Committee Title Software Engineering Secretariat: Standards Council of Canada (SCC) | Circulated to P-members of the committee for voting Please return all votes and comments in electronic form directly to the SC 7 Secretariat by the due date indicated. |
|---|--|

ISO/IEC JTC1/SC7

Title: Information Technology - Software Engineering Data Definition and Interchange - Transfer Part 3: Encoding ENCODING.1

Project: 07.28.2.3

Vote:

- APPROVAL OF THE DRAFT AS PRESENTED
- APPROVAL OF THE DRAFT WITH COMMENTS AS GIVEN ON THE ATTACHED
 - general:
 - technical:
 - editorial:
- DISAPPROVAL OF THE DRAFT FOR REASONS ON THE ATTACHED
 - Acceptance of these reasons and appropriate changes in the text will change our vote to approval
- ABSTENTION (FOR REASONS BELOW):

P-member voting:
National Body (Acronym)

Date:
YYCC-MM-DD

Submitted by:

Name



SC7

N1544

ISO IEC CD 15475-3

EIA/IS-110

SC7 WG11

N042R1

**Information Technology - Software Engineering Data
Definition and Interchange - Transfer Format
Part 3: Encoding ENCODING.1**

Author: EIA CDIF
Project: 07.28.2.3
Action: Circulation to JTC1 SC7
For CD Ballot, CD Review and
Comments
Source: SC7 WG11
Date: 1996.05.31
Status: Committee Draft

Reference: SC7 N1220 Resolution #343 Ottawa Plenary (94.06.17)
SC7 N1313 CDIF ENCODING.1 Working Draft
SC7 N1415 Resolution #404 Brisbane Plenary (96.06.01)

Convenor: Peter L. Eirich
Information Sciences Dept
Applied Research Lab.
Pennsylvania State Un.
P.O. Box 30 -- Room 210A.
State College, PA
USA 16804-0030

Tel: 1 (814) 865-3739
Fax: 1 (814) 863 1396
ple2c@psu.edu
eirich@erols.com

Secretary: James A. Williamson
Naval Air Warfare Center
Code 4572 / MS 85,
Warminster, Pennsylvania,
USA 18974

Tel: 1 215 441 1564
Fax: 1 215 441 1424
jaw@nadc.navy.mil

Introduction

ISO/IEC JTC1 SC7 WG11, in its Project 7.28, has the objective of enabling the interchange of information between a variety of heterogeneous software engineering tools.

Many of the input documents for Project 7.28 are the result of developmental work performed by the CASE Data Interchange Format (CDIF) technical committee of the Electronics Industries Association (EIA). This is one of those documents, and is distributed as received from the CDIF editor.

At its Brisbane meeting, in June 95, SC7 approved the distribution of the committee drafts.

This document is intended to provide one part within the multipart standard resulting from Project 7.28. Documents will be reformatted to ISO Directives as part of the Committee Draft ballot and comment resolution.

Comments

Comments are requested within the timetable provided by the SC7 cover sheet (generally 100 days) to facilitate preparation for the next WG11 meeting. Comments should be sent by normal formal channels.

To facilitate planning and preparatory resolution proposals, WG11 would appreciate comments to be submitted to the WWW (www.cdif.org/comments.html) site identified for that purpose).

Comments are to be provided in electronic format, generally following the format used for the WWW page. If comments are also submitted on paper, they should use a similar format.

Please identify clearly the status of comments submitted, such as:

National Body Comments,
National Body Approved Expert Comments.

Jean Bérubé
Project Editor, 7.28

IDEgenic inc
87 Burton,
Greenfield Park, Québec,
Canada J4V 2Y6

Tél: 1 514 465 0245
Fax: 1 514 465 2398
jberube@ibm.net

711PRA-11
Explanatory Notes for SC7 Reviewers
of the
WG11-CDIF Documents
(Project 07.28)

- The “Overview” document (WG11 N047R1, SC7 N1540) should be read first since it describes all the other project 7.28 CD documents to be reviewed at this time (as well as several documents planned for the future). It also describes the overall architecture that applies to these documents.
- The “Framework” document (WG11 N046R1, SC7 N1541) should be read second, as it describes the entity-relationship-attribute modeling notation that is used in many of the documents.
- The “Data Modeling” document (WG11 N036, SC7 N1548) and the “Data Flow Modeling” (WG11 N035, SC7 N1549) document provide the substantive coverage of software engineering techniques within the first set of documents. These documents permit the structured representation of software designs prepared by the use of any of a number of well established software engineering techniques that produce what are commonly called data flow diagrams and/or entity-relationship- attribute diagrams. These two documents will be of the greatest interest to software engineering practitioners.
- The “Foundation”, “Common”, and “Data Definition” documents (WG11 N048R1, N049, N050 and respectively SC7 N1545, N1546, 1547) are “modeling infrastructure” documents that support the contents of the software engineering techniques document, Data Modeling and Data Flow Modeling. Elements in these three infrastructure documents are incorporated by reference into the technique documents as necessary.
- The Transfer Format documents, “General Rules”, “Syntax”, and “Encoding” (WG11 N040R1, N041R1, N042R1 and respectively SC7 N1542, N1543, N1544) describe how character sequences are formed in order to communicate software design representations. These may be read separately from the technique and infrastructure documents described above.
- Of special importance for this review is the “Foundation” document (WG11 N048R1, SC7 N1545). This document illustrates the format intended to be applied to all the documents in this set for subsequent balloting. WG11 specifically solicits reviewer comments related to the format of this document so that the reformatting effort for the other documents in this set need be done only once.

WG11 will greatly appreciate the national body review comments on these documents, and thanks in advance the reviewers for their efforts. Any early submissions of comments will be very helpful.

In order to fully ensure that all comments received are processed efficiently and consistently, WG11 requests of all reviewers to please utilize the comments template that is provided with each document. Submitting comments in electronic form rather than on paper will be especially helpful.

If assistance with the usage of the comment forms is required, or if there are any other questions, please contact either Jean Berube, the Project 7.28 coordinating editor, or myself.

Sincerely,

Peter Eirich
WG11 Convenor

CD 15475-3

**Information Technology - Software Engineering Data
Definition and Interchange - Transfer Format**

Part 3: Encoding ENCODING.1

CDIF TITLE:

CDIF - Transfer Format - Encoding ENCODING.1

EIA/IS-110 JANUARY, 1994

Revised 4/18/96 to incorporate ISO/IEC JTC1 SC7/WG11 comments submitted as input to the 1995 Brisbane meeting.

ELECTRONIC INDUSTRIES ASSOCIATION

ENGINEERING DEPARTMENT

NOTICE

EIA Engineering standards and publications are designed to serve the public interest through eliminating misunderstandings between manufacturers and purchasers, facilitating interchangeability and improvement of products, and assisting the purchaser in selecting and obtaining with minimum delay the proper product for his particular need. Existence of such standards and publications shall not in any respect preclude any member or nonmember of EIA from manufacturing or selling products not conforming to such standards and publications, nor shall the existence of such standards and publications preclude their voluntary use by those other than EIA members, whether the standard is to be used either domestically or internationally.

Recommended standards and publications are adopted by EIA without regard to whether or not their adoption may involve patents on articles, materials or processes. By such action, EIA does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the standard or publication.

EIA INTERIM STANDARDS

EIA Interim Standards contain information deemed to be of technical value to the industry, and are published at the request of the originating Committee without necessarily following the rigorous public review and resolution of comments which is a procedural part of the development of an EIA Recommended Standard.

EIA Interim Standards should be reviewed on an annual basis by the formulating Committee and a decision made on whether to proceed to develop an EIA Recommended Standard on this subject. EIA Interim Standards must be canceled by the Committee and removed from the EIA Standards Catalog before the end of their third year of existence.

This EIA standard is considered to have international standardization implication, but the International Standardization Organization activity has not progressed to the point where a valid comparison between the EIA standard and the ISO work can be made.

COPYRIGHT 1994

Published by
ELECTRONIC INDUSTRIES ASSOCIATION
Engineering Department
2500 Wilson Blvd. Suite 203
Arlington, VA 22201

PRICE: Please refer to the current
Catalog of EIA & JEDEC STANDARDS & ENGINEERING PUBLICATIONS
or contact the EIA Electronic Information Group
(Phone: (703) 527-7001, Fax: (703) 527-0685)

All rights reserved
Printed in U.S.A.

ISBN 0-7908-0016-0

PREFACE

This standard will assist the vendors and users of CASE tools in developing mechanisms for interchanging information between CASE tools. This standard specifies an element of a family of related standards. When used together, these standards specify a mechanism for transferring information between CASE tools.

CDIF - CASE Data Interchange Format - Overview (ISO CD 15474-1: Information Technology - Software Engineering Data Definition and Interchange - Overview and Framework: Part 1 : Overview) and CDIF - Framework for Modeling and Extensibility (ISO CD 15474-2: Information Technology - Software Engineering Data Definition and Interchange - Overview and Framework: Part 2 : Framework for Modeling and Extensibility) should be read first when initially exploring CDIF. The first explains the overall CDIF Architecture and how the family of standards fits together. The second explains the scope, and modeling approach in CDIF. The CDIF Meta-meta-model and extensibility mechanism are also defined in that document.

This standard defines an encoding of SYNTAX.1. *EIA/IS-108 CDIF - Transfer Format - General Rules for Syntaxes and Encodings* describes how CDIF supports multiple syntaxes and encodings. *EIA/IS-109 CDIF - Transfer Format - Syntax SYNTAX.1* defines the initial CDIF Transfer Format syntax, SYNTAX.1.

This standard has been developed with the wide support and participation of vendors, users, academia and government involved in or familiar with the CASE industry, its products and the general requirements associated with interchanging information between these products.

The material contained in this publication has been copyrighted by the CASE Data Interchange Format (CDIF) Division of the Electronic Industries Association (EIA). Permission to copy this material in paper and electronic form has been granted to the international software engineering standards development organization (ISO/IEC JTC1/SC7) for the sole purpose of international standards development work. Organizations receiving copies of this document may not duplicate and distribute additional copies internally for any reason other than to facilitate their organization's review and comment back to ISO/IEC JTC1/SC7.

This page is intentionally blank

CONTENTS

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38

| | |
|--|-----------|
| 1. SCOPE | 1 |
| 2. INTRODUCTION..... | 2 |
| 2.1 Audience | 2 |
| 2.2 Structure of this Document..... | 2 |
| 2.3 Conventions | 4 |
| 2.3.1 Global Conventions | 4 |
| 2.3.2 BNF Conventions | 4 |
| 3. CONCEPTS AND DEFINITIONS..... | 6 |
| 3.1 Encoding Identifier | 6 |
| 3.2 Clear Text..... | 6 |
| 3.3 Character Sets..... | 6 |
| 4. ENCODING STRUCTURES AND KEYWORDS IN THE CDIF TRANSFER..... | 7 |
| 4.1 Introduction..... | 7 |
| 4.2 Encoding Structures..... | 7 |
| 4.2.1 Decimal Integer Value | 7 |
| 4.2.2 Binary Value..... | 7 |
| 4.2.3 Octal Value..... | 8 |
| 4.2.4 Hexadecimal Value..... | 8 |
| 4.2.5 Float Value | 8 |
| 4.2.6 Date 9 | |
| 4.2.7 Time 10 | |
| 4.2.8 Identifier Value..... | 10 |
| 4.2.9 Enumerated Value | 11 |
| 4.2.10 String..... | 11 |
| 4.2.11 Text String..... | 12 |
| 4.2.12 Comment | 13 |
| 4.2.13 MetaMetaObjectName | 14 |
| 4.2.14 MetaObjectName | 15 |
| 4.2.15 Identifier..... | 15 |
| 4.2.16 MultibyteIdentifier..... | 16 |
| 4.3 Keywords..... | 16 |
| 5. ENCODING FORMAL GRAMMAR..... | 17 |
| APPENDIX A TRANSFER EXAMPLES..... | 24 |
| A1. Introduction..... | 24 |
| A2. Example 1 | 24 |
| A2.1 Introduction | 24 |
| A2.2 Layout and Content..... | 24 |

| | | |
|----|---|--------------------------|
| 1 | A2.3 The Transfer | 25 |
| 2 | A3. Example 2 | 30 |
| 3 | A3.1 Introduction | 30 |
| 4 | A3.2 Overview | 30 |
| 5 | A3.3 SubjectArea Definition | 31 |
| 6 | A3.4 Definition of the Additions | 32 |
| 7 | A3.4.1 MetaEntity Definitions | 33 |
| 8 | A3.4.1.1 SecurityClassification | 33 |
| 9 | A3.4.2 MetaRelationship Definitions..... | 35 |
| 10 | A3.4.2.1 | |
| 11 | SemanticInformationObject.IsControlledBy. | |
| 12 | SecurityClassification | 35 |
| 13 | A3.5 Definition of the Extensions..... | 36 |
| 14 | A3.6 The Transfer | 37 |
| 15 | APPENDIX B QUESTIONS AND ANSWERS | 47 |
| 16 | GLOSSARY | 49 |
| 17 | INDEX..... | 59 |
| 18 | RELATED EIA DOCUMENTS | INSIDE BACK COVER |

FIGURES

1
2
3
4

Figure 1 Position in the CDIF Family of Standards 1

Figure A1 Semantic Objects in Example 25

This page is intentionally blank

1. SCOPE

The CDIF Family of Standards is primarily designed to be used as a description of a mechanism for transferring information between CASE tools. It facilitates a successful transfer when the authors of the importing and exporting tools have nothing in common except an agreement to conform to CDIF. The language that is defined for the Transfer Format also has applicability as a general language for Import/Export from repositories. The CDIF Integrated Meta-model defined for CASE also has applicability as the basis of standard definitions for use in repositories.

The standards which form the complete family of CDIF Standards are documented in *EIA/IS-106 CDIF - CASE Data Interchange Format - Overview*. These standards cover the overall framework, the transfer format and the CDIF Integrated Meta-model.

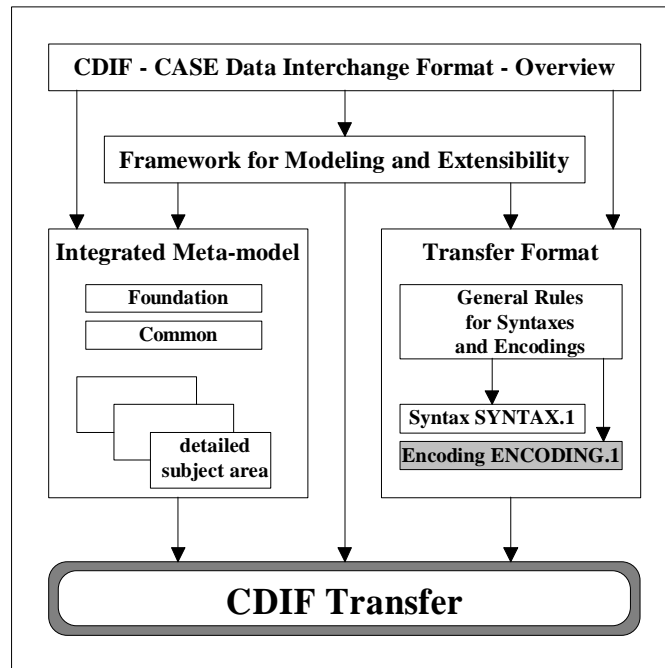


Figure 1
Position in the CDIF Family of Standards

The diagram in Figure 1 depicts the various standards that comprise the CDIF Family of Standards. The shaded box depicts this Standard and its position in the CDIF Family of Standards.

This document describes the standard CDIF Transfer Encoding for the standard CDIF Transfer Syntax as defined in *EIA/IS-109 CDIF - Transfer Format - Syntax SYNTAX.1*.

2. INTRODUCTION

2.1 Audience

This document is intended to be used by anyone wishing to understand and/or use CDIF. This document provides a definition of an encoding for CDIF transfers. It is suitable for:

- those evaluating CDIF
- those who wish to understand the principles and concepts of a CDIF transfer
- those developing importers and exporters.

The document *EIA/IS-106 CDIF - CASE Data Interchange Format - Overview* and the document *EIA/IS-107 CDIF - Framework for Modeling and Extensibility* should be read first when initially exploring CDIF and before attempting to read other documents in the CDIF Family of Standards.

This document should be read in conjunction with *EIA/IS-108 CDIF - Transfer Format - General Rules for Syntaxes and Encodings* and *EIA/IS-109 CDIF - Transfer Format - Syntax SYNTAX.1*.

While there are no specific prerequisites for reading this document, it will be helpful for the reader to have familiarity with the following:

- Entity-Relationship-Attribute modeling
- CASE tools
- Information repositories
- Data dictionaries
- Multiple meta-layer modeling
- Formal grammars
- Transfer Formats.

2.2 Structure of this Document

This document is organized into the following sections:

Section 1 'Scope'

This describes the CDIF Family of Standards and positions the current document within the set of documents.

1 **Section 2 'Introduction'**

2 This describes the intended audience, structure and conventions used in the document.

3 **Section 3 'Concepts and Definitions'**

4 This specifies the unique identifier for the encoding defined in this document and describes
5 the concepts of clear text and character sets used in this document.

6 **Section 4 'Encoding Structures and Keywords in the CDIF Transfer'**

7 This section describes and gives examples of the format of transfer syntax terminals and
8 lists the encodings of keywords.

9 **Section 5 'Encoding Formal Grammar'**

10 This section defines the encoding rules for all non-terminals in the encoding.

11 **Appendix A 'Transfer Examples'**

12 This gives two examples of a complete transfer, one with, and one without, extensibility.

13 **Appendix B 'Questions and Answers'**

14 This includes detailed responses to questions that have been raised by those reviewing this
15 document. It is intended that the inclusion of these will aid the reader in understanding
16 this document, and related issues concerning CDIF.

17 **Glossary**

18 The Glossary contains definitions of terms used within this Standard.

19 **References**

20 This contains details of other standards referenced within this Standard.

21 **Index**

22 A comprehensive index is provided.

2.3 Conventions

2.3.1 Global Conventions

The typographical and naming conventions defined below are used throughout the CDIF Family of Standards.

- Double quotes are used to introduce new terms (e.g., "model layer")
- Bold type is used for emphasis (e.g., this is **important**)
- All meta-objects and meta-meta-objects in CDIF (in meta-models and meta-meta-models) are named by concatenating all the words that name the meta-object or meta-meta-object; the first letter of each word is upper-case, the rest are lower-case (e.g., *MetaAttribute*, *AttributeDerivation*, *IsDrawnUsing*, *IsOptional*)

2.3.2 BNF Conventions

The grammar in this document is defined using an extended version of "Backus Naur Form" (BNF).

In BNF, each syntactic element of the language is defined by means of a "production rule". A "production rule" defines a syntactic element in terms of an expression, ordered left to right, that can be used to form an instance of the element. The expression can consist of characters, character strings, and other syntactic elements.

The version of BNF used in this document has the following conventions:

| <i>Symbol</i> | <i>Meaning</i> |
|---------------|---|
| < > | Angle braces delimit character strings that are the names of syntactic elements, i.e., non-terminal symbols. |
| ::= | This is the definition operator. It is used in a production rule to separate the element defined by the rule from its definition. The element being defined is to the left of the operator and the expression that defines the element is to the right of the operator. |
| [] | Square braces enclose optional elements in an expression. The portion of the expression within the braces may be explicitly specified or may be omitted. |
| { } | Curly braces group elements in an expression, so that the group can be treated as a single syntactic element. |

| | | |
|----|-----------------|--|
| 1 | | This is the alternative operator. The vertical bar indicates that the |
| 2 | | portion of the expression to the right of the bar is an alternative to the |
| 3 | | portion to the left of the bar. If the vertical bar appears at a position |
| 4 | | where it is not enclosed in curly or square braces, it specifies a |
| 5 | | complete alternative for the element defined by the production rule. If |
| 6 | | the vertical bar appears in a portion of the expression enclosed in curly |
| 7 | | or square braces, it specifies alternatives for the contents of the |
| 8 | | innermost pair of such braces. |
| 9 | ... | The ellipsis indicates that the element to which it applies in a formula |
| 10 | | may be repeated any number of times (but note that optionality is |
| 11 | | covered by the square braces convention). If the ellipsis appears |
| 12 | | immediately to the right of a closing curly or square brace, "}" or "]", |
| 13 | | then it applies to the portion of the expression enclosed between that |
| 14 | | closing brace and the corresponding opening brace, "{" or "[". If an |
| 15 | | ellipsis appears immediately to the right of any other element, then it |
| 16 | | applies only to that element. |
| 17 | !! | This is the comment operator. This operator introduces normal English |
| 18 | | text. It is used as a comment in the BNF and when the definition of a |
| 19 | | syntactic element is not expressed further in BNF. |
| 20 | <TokenName> | Anything emboldened and surrounded by angle braces is, a token , a |
| 21 | | BNF non-terminal symbol that is not further defined in the syntax. The |
| 22 | | detailed representation of a token is specified in the encoding. |
| 23 | Terminal | Anything emboldened, and not surrounded by angle braces, is a literal |
| 24 | | value (i.e., it represents the character string itself). |
| 25 | x .. y | This is the range operator. It refers to any symbol in the range x to y. |
| 26 | | For example, 0..255 refers to all integers 0 to 255 inclusive. |

3. CONCEPTS AND DEFINITIONS

3.1 Encoding Identifier

The encoding defined in this version of this standard shall have a CDIF Standardized Syntax Identifier of "ENCODING.1" and a Version of "02.00.00". This is used to define the standardized encoding used in a CDIF Transfer that conforms to the CDIF Standardized Syntax with Identifier "SYNTAX.1" and Version "02.00.00" (see *EIA/IS-109 CDIF - Transfer Format - Syntax SYNTAX.1*).

The syntax of the CDIF Encoding ID and Encoding Version is as follows:

```
<EncodingID> ::= "ENCODING.1"
```

```
<EncodingVersion> ::= "02.00.00"
```

3.2 Clear Text

The objectives of this clear text encoding are that it should be:

- Machine writeable: A transfer file using such an encoding should be capable of being generated by computer software.
- Machine readable: A transfer file using such an encoding should be capable of being parsed by computer software.
- Human readable: A transfer file using such an encoding should be capable of being read and understood by a human who is familiar with the concepts being transferred.

CDIF transfer files are not designed to be created or edited by humans, although it is not difficult to do so.

3.3 Character Sets

Whitespace characters are used (outside strings, text strings and comments) to separate tokens and to assist in formatting the transfer to improve its legibility. After token identification, these characters are ignored by importers. The whitespace characters are space, horizontal tab, vertical tab, carriage return, line feed and form feed.

Whitespace characters and all other characters in a transfer file are encoded using the character set specified in the Transfer Envelope.

Whitespace characters must not appear anywhere other than as defined in the syntax defined in *EIA/IS-109 CDIF - Transfer Format - Syntax SYNTAX.1*. This does not mean that a text string can not have non-printable characters. For more details, see section 4.2.11 Text String.

4. ENCODING STRUCTURES AND KEYWORDS IN THE CDIF TRANSFER

4.1 Introduction

A transfer expressed in this clear text encoding consists of a stream of characters forming a sequence of tokens separated by whitespace. The rules for separation of tokens are given in *EIA/IS-109 CDIF - Transfer Format - Syntax SYNTAX.1*.

4.2 Encoding Structures

4.2.1 Decimal Integer Value

A Decimal Integer is defined as follows:

```
<DecimalIntegerValue> ::= <DecimalRadix> <Integer>
<DecimalRadix> ::= #D
<Integer> ::= [ <Sign> ] <Digit> ...
```

The following are examples of valid and invalid Decimal Integer Values:

- (1) #d12345 - Valid: positive decimal
- (2) #d-12345 - Valid: negative decimal
- (3) #d+12345 - Valid: positive decimal
- (4) #d123e43 - Invalid: 'e' is not a numeric digit

4.2.2 Binary Value

A Binary Value is defined as follows:

```
<BinaryValue> ::= <BinaryRadix> <BinaryNumber>
<BinaryRadix> ::= #B
<BinaryNumber> ::= [ <Sign> ] <BinaryDigit> ...
<BinaryDigit> ::= 0 | 1
```

The following are examples of valid and invalid Binary Values:

- (1) #b10101 - Valid: positive binary
- (2) #b-10101 - Valid: negative binary
- (3) #b11021 - Invalid: '2' is not a binary digit

4.2.3 Octal Value

An Octal Value is defined as follows:

```
<OctalValue> ::=          <OctalRadix> <OctalNumber>
<OctalRadix> ::=          #O
<OctalNumber> ::=         [ <Sign> ] <OctalDigit> ...
<OctalDigit> ::=         0 | 1 | 2 | 3 | 4 | 5 | 6 | 7
```

The following are examples of valid and invalid Octal values:

- (1) #o31727 - Valid: positive octal
- (2) #o-31727 - Valid: negative octal
- (3) #o31923 - Invalid: '9' is not an octal digit

4.2.4 Hexadecimal Value

A Hexadecimal Value is defined as follows:

```
<HexadecimalValue> ::=    <HexRadix> <HexNumber>
<HexRadix> ::=            #H
<HexNumber> ::=          [ <Sign> ] <HexDigit> ...
<HexDigit> ::=           0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
                          | A | B | C | D | E | F
                          | a | b | c | d | e | f
```

The following are examples of valid and invalid Hexadecimal values:

- (1) #h1e2cf - Valid: positive hexadecimal
- (2) #hffffffff - Valid: positive hexadecimal
- (3) #-hffffffff - Valid: negative hexadecimal
- (4) #h122g9 - Invalid: 'g' not a valid hex digit

4.2.5 Float Value

A Float Value is a decimal number with up to 16 decimal digits of precision within the range of 10^{-1023} to 10^{1023} .

1 A Float Value is defined as follows:

```
2 <FloatValue> ::= <FloatDesignator> <Mantissa> <Exponent>
3 <FloatDesignator> ::= #F
4 <Mantissa> ::= [ <Sign> ] <Digit> ...
5 [ <DecimalPoint> [ <Digit> ... ] ]
6 <DecimalPoint> ::= .
7 <Exponent> ::= <Exp> <Integer>
8 <Exp> ::= E
```

9 Examples of valid and invalid Float values are:

```
10 (1) #f4E9 - Valid
11 (2) #f123.45E2 - Valid
12 (3) #f-123.45E15 - Valid
13 (4) #f+0.23E5 - Valid
14 (5) #f0.23E-3 - Valid
15 (6) #f0.23E+3 - Valid
16 (7) #f123.45 - Invalid: missing exponent
17 (8) #f.23E3 - Invalid: must follow radix with a sign or number
18 (9) #f23.32-2 - Invalid: missing 'E' exponent prefix
```

20 4.2.6 Date

21 A Date is defined as follows:

```
22 <Date> ::= <YearValue> <DateDelimiter>
23 <MonthValue> <DateDelimiter>
24 <DayValue>
25 <YearValue> ::= <Integer>
26 <MonthValue> ::= <Integer>
27 <DayValue> ::= <Integer>
28 <DateDelimiter> ::= /
```

1 The following are examples of valid and invalid date values:

- | | | | | |
|---|-----|-------------|---|--|
| 2 | (1) | 1940/12/07 | - | Valid |
| 3 | (2) | 0002/10/11 | - | Valid |
| 4 | (3) | 1941/Dec/12 | - | Invalid: month must be an integer |
| 5 | (4) | 1991/11/31 | - | Invalid: day out of valid range for month. |

6 7 **4.2.7 Time**

8 A Time is defined as follows:

```
9 <Time> ::= <Hours> <TimeDelimiter>
10 <Minutes> <TimeDelimiter>
11 <Seconds>
12
12 <Hours> ::= <Integer>
13
13 <Minutes> ::= <Integer>
14
14 <Seconds> ::= <Integer>
15 [ <DecimalPoint> <Digit>
16 [ <Digit> [ <Digit> ] ] ]
17
17 <TimeDelimiter> ::= :
```

18 The following are examples of valid and invalid time values:

- | | | | | |
|----|-----|--------------|---|--------------------------------|
| 19 | (1) | 07:20:23 | - | Valid |
| 20 | (2) | 03:59:59 | - | Valid |
| 21 | (3) | 00:00:00.250 | - | Valid |
| 22 | (4) | 29:25:19 | - | Invalid: illegal hour value 29 |

23 24 **4.2.8 Identifier Value**

25 The syntax of the Identifier Value clause is:

```
26 <IdentifierValue> ::= * <MultibyteIdentifier> *
```

27 <MultibyteIdentifier> must be a member of the list of values given in the definition of the
28 meta-attribute.

29 The following are examples of valid identifier values:

- | | | | | |
|----|-----|-------------------|---|-------|
| 30 | (1) | *johnBrownsBody* | - | Valid |
| 31 | (2) | *sysrange* | - | Valid |
| 32 | (3) | *CUSTOMER-RECORD* | - | Valid |

1 (4) *9system* - Valid
2

3 4.2.9 Enumerated Value

4 The syntax of the Enumerated Value clause is:

5 <EnumeratedValue> ::= < <MultibyteIdentifier> >

6 <MultibyteIdentifier> must be a member of the list of values given in the definition of the
7 meta-attribute.

8 The following are examples of valid enumerated values:

9 (1) <red> - Valid
10 (2) <10> - Valid
11

12 4.2.10 String

13 A String is defined as follows:

14 <String> ::= <StringDelimiter> [<StringElement>] ...
15 <StringDelimiter>
16 <StringDelimiter> ::= "
17 <StringElement> ::= <GeneralPrintableChar>
18 |]
19 | |
20 | #
21 | <EscapeCharacter> <StringDelimiter>
22 | <EscapeCharacter> <EscapeCharacter>

23 If it is necessary to include a pattern in a string that matches the <StringDelimiter>, then this
24 must be "escaped" by preceding each occurrence of the pattern by the <EscapeCharacter>. In
25 order to embed a pattern that is the same as the <EscapeCharacter>, the pattern must be
26 preceded by the <EscapeCharacter>.], |, and # are explicitly listed as <StringElement>s
27 because they are not <GeneralPrintableChar>s. <GeneralPrintableChar> does not include
28 characters like ",], |, and # because they are used as a delimiter (or part of a delimiter) in strings,
29 text strings, and comments.

30 The following are examples of valid and invalid strings:

31 (1) "This is a string" - Valid
32 (2) "This is a "string" - Invalid: embedded delimiter not escaped.
33 (3) "This is a \"string\"" - Valid: embedded delimiters escaped.

1 (4) "Here is an escape character \\ (backslash) in a string"
2 - Valid
3

4 **4.2.11 Text String**

5 A Text String represents a sequence of printable characters, which may include embedded
6 "whitespace" characters. It is defined as follows:

7 <TextString> ::= <OpenText> [<TextElement>] ... []]
8 <CloseText>

9 <OpenText> ::= # [

10 <TextElement> ::= <GeneralPrintableChar>
11 | "
12 |
13 | #
14 | <WhiteSpace>
15 | <EscapeCharacter> <CloseText>
16 | <EscapeCharacter> <EscapeCharacter>
17 |] <TextElementExcludingHashSign>
18 | <OtherChar>

19 <TextElementExcludingHashSign> ::= <GeneralPrintableChar>
20 | "
21 |
22 | <WhiteSpace>
23 | <EscapeCharacter> <CloseText>
24 | <EscapeCharacter> <EscapeCharacter>
25 |] <TextElementExcludingHashSign>
26 | <OtherChar>
27

28 <OtherChar> ::= <EscapeCharacter> <HexadecimalValue>
29 <EscapeCharacter>

30 <CloseText> ::=] #

31 The maximum size of a <TextString> is 1024 characters between the <OpenText> and
32 <CloseText> delimiters.

33 If it is necessary to include a pattern in the text string that matches the <CloseText> delimiter,
34 then this must be "escaped" by preceding each occurrence of the pattern by the
35 <EscapeCharacter>. In order to embed a pattern that is the same as the <EscapeCharacter>,
36 the pattern must be preceded by the <EscapeCharacter>. ",], |, and # are explicitly listed as
37 <TextElement>s because they are not <GeneralPrintableChar>s.
38 <GeneralPrintableChar> does not include characters like ",], |, and # because they are used
39 as a delimiter (or part of a delimiter) in strings, text strings, and comments.

1 In order to embed a character that is "not printable" (it is not a <GeneralPrintableChar> , ", #,
2], <EscapeCharacter> or <WhiteSpace>), use <OtherChar>. <OtherChar> encodes a
3 character as an escaped sequence with an embedded hexadecimal number. This hexadecimal
4 number is the hexadecimal equivalent of the binary number used to represent the character being
5 transferred.

6 The following are examples of valid and invalid text strings¹:

```
7 (1)  #[Program SumIntegers(Input,Output);  
8     var  
9     total,input_integer : Integer;  
10    begin  
11    while not EOF(Input) do  
12        begin  
13        ReadLn(input_integer);  
14        total:=total+input_integer  
15        end;  
16    WriteLn('Total =',total);  
17    end.]#           -       Valid  
18 (2)  #[This is some text]#   -       Invalid: incomplete text delimiter  
19 (3)  #[The end of text marker is "]"#"]#  
20        -       Invalid: ]# not escaped  
21 (4)  #[Here is how to embed a \]# in the text]#  
22        -       Valid  
23 (5)  #[And here is how to embed two backslashes \\ in the text]#  
24        -       Valid  
25 (6)  #[This is some text]   -       Invalid: missing # at the end  
26 (7)  #[Control-Z is usually the character \#H1A\, which is not printable.]#  
27        -       Valid  
28 (8)  #[There is a right square bracket, ], embedded in this text.]#  
29        -       Valid  
30 (9)  #[There is a right square bracket, \], embedded (and escaped) in this  
31    text.]#           -       Invalid: \] not followed by #
```

33 4.2.12 Comment

34 A Comment is defined as follows:

35 <Comment> ::= <OpenComment> <CommentBody> <CloseComment>

36 <OpenComment> ::= #|

37 <CommentBody> ::= [<CommentBodyChar>] ... [|]

¹These examples of the TextString clause have been laid out with whitespace characters, such as <CR>, transformed for presentation. <eof> is used to represent the end of the file.

```
1      <CommentBodyChar> ::=          <WhiteSpace>
2                                  | <GeneralPrintableChar>
3                                  | "
4                                  | ]
5                                  | #
6                                  | <EscapeCharacter> <CloseComment>
7                                  | <EscapeCharacter> <EscapeCharacter>
8                                  | <CommentBodyCharExcludingHashSign>
9
9      <CommentBodyCharExcludingHashSign> ::=
10                                 <WhiteSpace>
11                                 | <GeneralPrintableChar>
12                                 | "
13                                 | ]
14                                 | <EscapeCharacter> <CloseComment>
15                                 | <EscapeCharacter> <EscapeCharacter>
16                                 | <CommentBodyCharExcludingHashSign>
17
17     <CloseComment> ::=           |#
```

18 ",], and # are explicitly listed as <CommentBodyChar>s because they are not
19 <GeneralPrintableChar>s. <GeneralPrintableChar> does not include characters like ",],
20 |, and # because they are used as a delimiter (or part of a delimiter) in strings, text strings, and
21 comments.

22 The following are examples of valid and invalid comments, represented in the standard CDIF
23 encoding²:

- ```
24 (1) #| this is
25 a multi-line comment |# - Valid
26 (2) (#d1 #d2 #| buckle my shoe |# #d3 #d4) - Valid (in line comment)
27 (3) #| hello ... <eof> - Invalid (missing delimiter)
```

### 29 4.2.13 MetaMetaObjectName

30 A Meta-meta-object Name is defined as follows:

```
31 <MetaMetaObjectName> ::= <UpperCaseAlphabeticChar>
32 [<UpperOrLowerCaseAlphabeticChar>] ...
```

---

<sup>2</sup>These examples of the Comment clause have been laid out with whitespace characters, such as <CR>, transformed for presentation.

1 The following are examples of valid and invalid MetaMetaObjectNames:

- 2 (1) CollectableMetaObject - Valid
- 3 (2) Meta-Entity - Invalid (- is not an alphabetic character)
- 4 (3) metaattribute - Invalid (does not start with a capital)

#### 5 6 **4.2.14 MetaObjectName**

7 A Meta-object Name is defined as follows:

8 <MetaObjectName> ::= <MultibyteIdentifier> ...

9 The following are examples of valid and invalid MetaObjectNames:

- 10 (1) DataModel - Valid
- 11 (2) Entity - Valid
- 12 (3) DataModelAttribute - Valid
- 13 (4) Security-Classification - Invalid (- is not an alphabetic character)
- 14 (5) role - Invalid (does not start with a capital)

#### 15 16 **4.2.15 Identifier**

17 An Identifier is defined as follows:

18 <Identifier> ::= { <UpperOrLowerCaseAlphabeticChar>  
19 | <Digit> }  
20 [ <UpperOrLowerCaseAlphabeticChar>  
21 | <Digit> | <UnderScore>  
22 | <Hyphen> ] ...

23 Examples of valid and invalid Identifier values:

- 24 (1) johnBrownsBody - Valid
- 25 (2) custRec.employeeBadgeId - Invalid: embedded dot
- 26 (3) sysrange - Valid
- 27 (4) CUSTOMER-RECORD - Valid
- 28 (5) 9system - Valid
- 29 (6) John Brown - Invalid: embedded space
- 30 (7) EMPLOYEE\_RECORD - Valid

## 4.2.16 MultibyteIdentifier

An MultibyteIdentifier is defined as follows:

```
<MultibyteIdentifier> ::= !! Printable characters defined in the
 character set specified by the Transfer
 Envelope, excluding '\', '"', '[', ']', '|',
 '#', '<', '>', '*', ':', '+', '-', '(', ')',
 ',', and space.
```

## 4.3 Keywords

A keyword is encoded as a sequence of alphabetic, numeric, underscore ('\_'), hyphen ('-') and colon (':') characters using the character set specified by the Transfer Envelope. Alphabetic characters in keywords are case insensitive. The following are the encodings of keywords:

```
<BitmapKeyword> ::= :BITMAP
<DateKeyword> ::= :DATE
<ExtendMetaAttributeKeyword> ::= :EXTENDMETA-ATTRIBUTE
<HeaderKeyword> ::= :HEADER
<HeightKeyword> ::= :HEIGHT
<IntegerListKeyword> ::= :INTEGERLIST
<MetaModelKeyword> ::= :META-MODEL
<ModelKeyword> ::= :MODEL
<PointKeyword> ::= :POINT
<PointListKeyword> ::= :POINTLIST
<SubjectAreaReferenceKeyword> ::= :SUBJECTAREAREFERENCE
<SummaryKeyword> ::= :SUMMARY
<TimeKeyword> ::= :TIME
<VersionNumberKeyword> ::= :VERSIONNUMBER
<WidthKeyword> ::= :WIDTH
```

## 5. ENCODING FORMAL GRAMMAR

This section specifies the encoding production rules. Some of the production rules were presented in Section 4 Encoding Structures and Keywords in the CDIF Transfer. This BNF alone is not a complete definition of ENCODING.1. There are additional definitions and restrictions expressed in Section 3 Concepts and Definitions and in the text in Section 4 Encoding Structures and Keywords in the CDIF Transfer.

The case of alphabetic characters in the encodings of <PrintableChar>, <UpperCaseAlphabeticChar> and <UpperOrLowerCaseAlphabeticChar> is significant, otherwise case is not significant.

The following are the encoding rules:

```
<Absolute> ::= ABSOLUTE
<AbsoluteUTC> ::= ABSOLUTEUTC
<AbsoluteLocal> ::= ABSOLUTELOCAL
<BinaryDigit> ::= 0 | 1
<BinaryNumber> ::= [<Sign>] <BinaryDigit> ...
<BinaryRadix> ::= #B
<BinaryValue> ::= <BinaryRadix> <BinaryNumber>
<BitmapKeyword> ::= :BITMAP
<CloseComment> ::= |#
<CloseScope> ::=)
<CloseText> ::=]#
<Comment> ::= <OpenComment> <CommentBody> <CloseComment>
<CommentBody> ::= [<CommentBodyChar>] ... [|]
```

---

```

1 <CommentBodyChar> ::= <WhiteSpace>
2 | <GeneralPrintableChar>
3 | "
4 |]
5 | #
6 | <EscapeCharacter> <CloseComment>
7 | <EscapeCharacter> <EscapeCharacter>
8 | <CommentBodyCharExcludingHashSign>

9 <CommentBodyCharExcludingHashSign> ::=
10 <WhiteSpace>
11 | <GeneralPrintableChar>
12 | "
13 |]
14 | <EscapeCharacter> <CloseComment>
15 | <EscapeCharacter> <EscapeCharacter>
16 | <CommentBodyCharExcludingHashSign>

17 <Date> ::=
18 <YearValue> <DateDelimiter>
19 <MonthValue> <DateDelimiter>
20 <DayValue>

21 <DateKeyword> ::= :DATE

22 <DateDelimiter> ::= /

23 <DayValue> ::= <Integer>

24 <DecimalIntegerValue> ::= <DecimalRadix> <Integer>

25 <DecimalPoint> ::= .

26 <DecimalRadix> ::= #D

27 <Digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

28 <Dot> ::= .

29 <EnumeratedSeparator> ::= ,

30 <EnumeratedValue> ::= < <MultibyteIdentifier> >

31 <EscapeCharacter> ::= \

32 <Exp> ::= E

33 <Exponent> ::= <Exp> <Integer>

34 <ExtendMetaAttributeKeyword> ::=
 :EXTENDMETA-ATTRIBUTE

```

---

```
1 <FalseValue> ::= -FALSE-
2 <FloatDesignator> ::= #F
3 <FloatValue> ::= <FloatDesignator> <Mantissa> <Exponent>
4 <GeneralPrintableChar> ::= !! Printable characters defined in the
5 character set specified by the Transfer
6 Envelope, excluding '\', '"', ']', '|' and
7 '#'.
8
9 !! Note that the characters \, ",], /, and # have been purposely
10 left out of <GeneralPrintableChar> because \ is used as an
11 escape character and the others are used as or in delimiters.
12
13 <HeaderKeyword> ::= :HEADER
14
15 <HeightKeyword> ::= :HEIGHT
16
17 <HexadecimalValue> ::= <HexRadix> <HexNumber>
18
19 <HexDigit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
20 | A | B | C | D | E | F
21 | a | b | c | d | e | f
22
23 <HexNumber> ::= [<Sign>] <HexDigit> ...
24
25 <HexRadix> ::= #H
26
27 <Hours> ::= <Integer>
28
29 <Hyphen> ::= -
30
31 <Identifier> ::= { <UpperOrLowerCaseAlphabeticChar>
32 | <Digit> }
33 [<UpperOrLowerCaseAlphabeticChar>
34 | <Digit> | <UnderScore>
35 | <Hyphen>] ...
36
37 <IdentifierValue> ::= * <MultibyteIdentifier> *
38
39 <Integer> ::= [<Sign>] <Digit> ...
40
41 <IntegerListKeyword> ::= :INTEGERLIST
42
43 <ListSeparator> ::= ,
44
45 <LowerCaseAlphabeticChar> ::=
46 a | b | c | d | e | f | g | h | i | j | k
47 | l | m | n | o | p | q | r | s | t | u
48 | v | w | x | y | z
```

```
1 <Mantissa> ::= [<Sign>] <Digit> ...
2 [<DecimalPoint> [<Digit> ...]]
3
4 <MetaModelKeyword> ::= :META-MODEL
5
6 <MetaMetaObjectName> ::= <UpperCaseAlphabeticChar>
7 [<UpperOrLowerCaseAlphabeticChar>] ...
8
9 <MetaObjectName> ::= <MultibyteIdentifier> ...
10
11 <Minutes> ::= <Integer>
12
13 <ModelKeyword> ::= :MODEL
14
15 <MonthValue> ::= <Integer>
16
17 <MultibyteIdentifier> ::= !! Printable characters defined in the
18 character set specified by the Transfer
19 Envelope, excluding '\', '"', '[', ']', '|',
20 '#', '<', '>', '*', ':', '+', '-', '(', ')',
21 ', ', and space.
22
23 <OctalDigit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7
24
25 <OctalNumber> ::= [<Sign>] <OctalDigit> ...
26
27 <OctalRadix> ::= #0
28
29 <OctalValue> ::= <OctalRadix> <OctalNumber>
30
31 <OpenComment> ::= #|
32
33 <OpenScope> ::= (
34
35 <OpenText> ::= #[
36
37 <OtherChar> ::= <EscapeCharacter> <HexaDecimalValue>
38 <EscapeCharacter>
39
40 <PixelIntensity> ::= 0 .. 255
41
42 <PixelSeparator> ::= ,
43
44 <PointKeyword> ::= :POINT
45
46 <PointListKeyword> ::= :POINTLIST
47
48 <PointSeparator> ::= <Space>
49
50 <PositiveInteger> ::= [+] <Digit> ...
```

```
1 <RelativeNegative> ::= RELATIVENEGATIVE
2 <RelativePositive> ::= RELATIVEPOSITIVE
3 <Seconds> ::= <Integer>
4 [<DecimalPoint> <Digit>
5 [<Digit> [<Digit>]]]
6 <Sign> ::= + | -
7 <Space> ::=
8 !! Space character as defined in the character set
9 specified by the Transfer Envelope.
10 <String> ::= <StringDelimiter> [<StringElement>] ...
11 <StringDelimiter>
12 <StringDelimiter> ::= "
13 <StringElement> ::= <GeneralPrintableChar>
14 |]
15 |
16 | #
17 | <EscapeCharacter> <StringDelimiter>
18 | <EscapeCharacter> <EscapeCharacter>
19 <SubjectAreaReferenceKeyword> ::=
20 :SUBJECTAREAREFERENCE
21 <SummaryKeyword> ::= :SUMMARY
22 <TextElement> ::= <GeneralPrintableChar>
23 | "
24 |
25 | #
26 | <WhiteSpace>
27 | <EscapeCharacter> <CloseText>
28 | <EscapeCharacter> <EscapeCharacter>
29 |] <TextElementExcludingHashSign>
30 | <OtherChar>
31 <TextElementExcludingHashSign> ::=
32 <GeneralPrintableChar>
33 | "
34 |
35 | <WhiteSpace>
36 | <EscapeCharacter> <CloseText>
37 | <EscapeCharacter> <EscapeCharacter>
38 |] <TextElementExcludingHashSign>
39 | <OtherChar>
```

```
1 <TextString> ::= <OpenText> [<TextElement>] ... []]
2 <CloseText>

3 <Time> ::= <Hours> <TimeDelimiter>
4 <Minutes> <TimeDelimiter>
5 <Seconds>

6 <TimeKeyword> ::= :TIME

7 <TimeDelimiter> ::= :

8 <TrueValue> ::= -TRUE-

9 <UnderScore> ::= _

10 <UpperCaseAlphabeticChar> ::=
11 A | B | C | D | E | F | G | H | I | J | K
12 | L | M | N | O | P | Q | R | S | T | U
13 | V | W | X | Y | Z

14 <UpperOrLowerCaseAlphabeticChar> ::=
15 <UpperCaseAlphabeticChar>
16 | <LowerCaseAlphabeticChar>

17 <VersionNumberKeyword> ::= :VERSIONNUMBER

18 <WhiteSpace> ::=
19 !! Space, horizontal tab, vertical tab, carriage
20 return, line feed or form feed character as defined
21 in the character set specified by the Transfer
22 Envelope.

23 <WidthKeyword> ::= :WIDTH

24 <YearValue> ::= <Integer>

25
```

1

This page is intentionally blank

## APPENDIX A TRANSFER EXAMPLES

### A1. Introduction

There are two examples in this appendix. The first is an example of the transfer of a Model using the CDIF Integrated CASE Meta-model. The second is an example of a transfer using an extended meta-model.

They are both informational examples for SYNTAX.1 and ENCODING.1 only. A normative reference for the definition of a meta-object in the CDIF Integrated Meta-model is any CDIF Integrated Meta-model Subject Area standard that uses that meta-object. The relevant subject areas are referenced in the meta-model section of each CDIF transfer.

### A2. Example 1

#### A2.1 Introduction

This is an example of a transfer using the standard CDIF Data Modeling and Data Definition Subject Areas.

#### A2.2 Layout and Content

Whitespace has been used in the formatting of this example to assist in the interpretation of the transfer. The transfer includes all the semantic objects shown in Figure A1 (which is an extract from the example in *EIA/IS-114 CDIF - Integrated Meta-model - Data Modeling Subject Area*).

---

<sup>1</sup>This appendix is not a formal part of the attached EIA Standard but is included for purposes of information only.

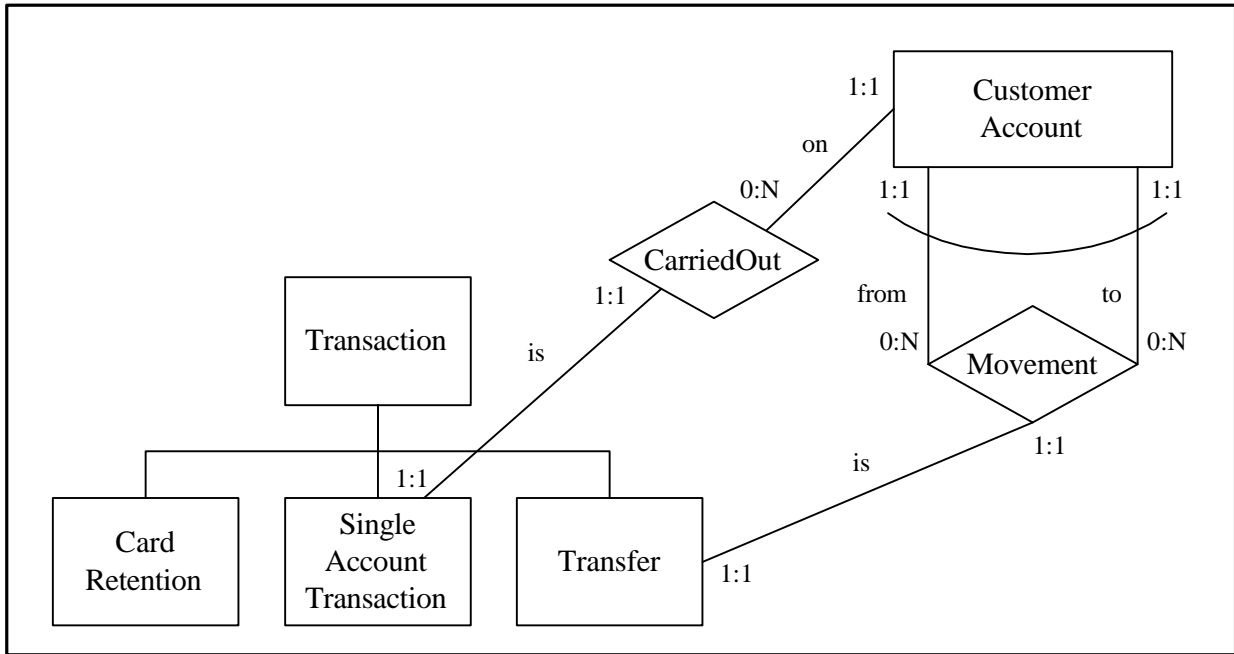


Figure A1  
 Semantic Objects in Example

### A2.3 The Transfer

```

1 CDIF,SYNTAX "SYNTAX.1" "02.00.00",ENCODING "ENCODING.1" "02.00.00", CODESET
2 "ISO-8859-1"
3 #| Sample CDIF Transfer using CDIF Integrated Meta-model |#
4 #| Header Section |#
5 (:HEADER
6 (:SUMMARY
7 (ExporterName "CASE Genius")
8 (ExporterVersion "01.00.00")
9 (ExportDate "1991/04/01")
10 (ExportTime "06:00:00")
11 (PublisherName "Mary Lomas")
12)
13)
14 #| Meta-model Section |#
15 (:META-MODEL
16 (:SUBJECTAREAREFERENCE DataModeling
17 (:VERSIONNUMBER "01.00")
18)
19 (:SUBJECTAREAREFERENCE DataDefinition
20 (:VERSIONNUMBER "01.00")
21)
22)
23)
24)

```

```
1)
2)
3 #| M o d e l S e c t i o n |#
4 (:MODEL
5 (DataModel MOD01
6 (Name "Example 1")
7 (BriefDescription "The first example Data Model.")
8 (ModelType "Conceptual")
9)
10 (Entity ENT02
11 (Name "Customer Account")
12 (BriefDescription "The bank account of a customer.")
13)
14 (Entity ENT06
15 (Name "Transaction")
16 (BriefDescription "An action generated by an ATM session against a
17 customer's bank account.")
18)
19 (Entity ENT07
20 (Name "Card Retention")
21 (BriefDescription "The action generated by retaining the Card
22 within the ATM.")
23)
24 (Entity ENT08
25 (Name "Single Account Transaction")
26 (BriefDescription "A subtype transaction indicating action against
27 one account.")
28)
29 (Entity ENT09
30 (Name "Transfer")
31 (BriefDescription "A subtype transaction indicating movement of
32 funds between multiple accounts.")
33)
34 (LocalAttribute DMATT12
35 (Name "Transaction Identifier")
36 (BriefDescription "The unique identifier of a Transaction.")
37 (IsOptional -False-)
38)
39 (LocalAttribute DMATT13
40 (Name "Transaction Date")
41 (BriefDescription "The date the transaction took place.")
42 (IsOptional -False-)
43)
44 (LocalAttribute DMATT14
45 (Name "Transaction Time")
46 (BriefDescription "The time the transaction took place.")
47 (IsOptional -False-)
48)
49 (LocalAttribute DMATT15
50 (Name "Customer Account Identifier")
51 (BriefDescription "The unique identifier of a Customer Account.")
```

```
1 (IsOptional -False-)
2)
3 (LocalAttribute DMATT16
4 (Name "Customer Account Type")
5 (BriefDescription "A code that indicates the type of account.")
6 (IsOptional -False-)
7)
8 (LocalAttribute DMATT17
9 (Name "Customer Account Balance")
10 (BriefDescription "The amount of money in a customer's account.")
11 (IsOptional -False-)
12)
13 (LocalAttribute DMATT20
14 (Name "Amount")
15 (BriefDescription "The amount of money transferred between
16 accounts.")
17 (IsOptional -False-)
18)
19 (Relationship REL04
20 (Name "CarriedOut")
21 (BriefDescription "This relates a Customer Account to the Single
22 Account Transaction performed against it.")
23)
24 (Relationship REL05
25 (Name "Movement")
26 (BriefDescription "This relates a Customer Account to a
27 Transfer.")
28)
29 (OrthogonalSubtypeSet OSS01
30 (Name "TransactionType")
31)
32 (Role ROLE09
33 (RoleName "On")
34 (MinOuterCardinality "1")
35 (MaxOuterCardinality "1")
36 (MinInnerCardinality "0")
37 (MaxInnerCardinality "N")
38)
39 (Role ROLE10
40 (RoleName "Is")
41 (MinOuterCardinality "1")
42 (MaxOuterCardinality "1")
43 (MinInnerCardinality "1")
44 (MaxInnerCardinality "1")
45)
46 (Role ROLE11
47 (RoleName "From")
48 (MinOuterCardinality "1")
49 (MaxOuterCardinality "1")
50 (MinInnerCardinality "0")
51 (MaxInnerCardinality "N")
```

```
1)
2 (Role ROLE12
3 (RoleName "To")
4 (MinOuterCardinality "1")
5 (MaxOuterCardinality "1")
6 (MinInnerCardinality "0")
7 (MaxInnerCardinality "N")
8)
9 (Role ROLE13
10 (RoleName "Is")
11 (MinOuterCardinality "1")
12 (MaxOuterCardinality "1")
13 (MinInnerCardinality "1")
14 (MaxInnerCardinality "1")
15)
16 (RolePlayer RP01
17 (Name "Actor")
18)
19 (RolePlayer RP02
20 (Name "Object")
21)
22 (RolePlayer RP03
23 (Name "Actor")
24)
25 (RolePlayer RP04
26 (Name "Sender")
27)
28 (RolePlayer RP05
29 (Name "Receiver")
30)
31 (IntegerType ELEM01
32 (Name "Int")
33 (SignedFlag -True-)
34)
35 (DataElementType TYPE01
36 (Name "Identifier")
37)
38 (DataElementType TYPE03
39 (Name "AccountType")
40)
41 (MoneyType TYPE04
42 (Name "Amount")
43)
44 (DateType TYPE08
45 (Name "Date")
46)
47 (TimeType TYPE09
48 (Name "Time")
49)
50 (DomainGroup DOM57)
51 (DomainValueEnumeration DOM58
```

```
1 (Name "Checking")
2)
3 (DomainValueEnumeration DOM59
4 (Name "Savings")
5)
6 (DataModel.IsCollectionOf.DataModelObject R001 MOD01 ENT02)
7 (DataModel.IsCollectionOf.DataModelObject R002 MOD01 ENT06)
8 (DataModel.IsCollectionOf.DataModelObject R003 MOD01 ENT07)
9 (DataModel.IsCollectionOf.DataModelObject R004 MOD01 ENT08)
10 (DataModel.IsCollectionOf.DataModelObject R005 MOD01 ENT09)
11 (DataModel.IsCollectionOf.DataModelObject R006 MOD01 REL04)
12 (DataModel.IsCollectionOf.DataModelObject R007 MOD01 REL05)
13 (InheritableDataModelObject.Has.OrthogonalSubtypeSet R044 ENT06 OSS01)
14 (OrthogonalSubtypeSet.IsConstructedWith.InheritableDataModelObject R008
15 OSS01 ENT07)
16 (OrthogonalSubtypeSet.IsConstructedWith.InheritableDataModelObject R009
17 OSS01 ENT08)
18 (OrthogonalSubtypeSet.IsConstructedWith.InheritableDataModelObject R010
19 OSS01 ENT09)
20 (DataModelObject.Plays.Role R011 ENT02 RP02)
21 (DataModelObject.Plays.Role R012 ENT08 RP01)
22 (DataModelObject.Plays.Role R013 ENT02 RP04)
23 (DataModelObject.Plays.Role R014 ENT02 RP05)
24 (DataModelObject.Plays.Role R015 ENT09 RP03)
25 (RolePlayer.Plays.Role R039 RP01 ROLE10)
26 (RolePlayer.Plays.Role R040 RP02 ROLE09)
27 (RolePlayer.Plays.Role R041 RP03 ROLE13)
28 (RolePlayer.Plays.Role R042 RP04 ROLE11)
29 (RolePlayer.Plays.Role R043 RP05 ROLE12)
30 (Role.BelongsTo.Relationship R016 ROLE09 REL04)
31 (Role.BelongsTo.Relationship R017 ROLE10 REL04)
32 (Role.BelongsTo.Relationship R018 ROLE11 REL05)
33 (Role.BelongsTo.Relationship R019 ROLE12 REL05)
34 (Role.BelongsTo.Relationship R020 ROLE13 REL05)
35 (DataType.HasSubtype.DataType R021 ELEM01 TYPE01)
36 (DataType.TakesValuesFrom.DomainGroup R022 TYPE03 DOM57)
37 (DomainGroup.Contains.Domain R023 DOM57 DOM58)
38 (DomainGroup.Contains.Domain R024 DOM57 DOM59)
39 (DataObject.IsDescribedBy.Attribute R025 ENT06 DMATT12
40 (SequenceNumber #d1)
41)
42 (DataObject.IsDescribedBy.Attribute R026 ENT06 DMATT13
43 (SequenceNumber #d2)
44)
45 (DataObject.IsDescribedBy.Attribute R027 ENT06 DMATT14
46 (SequenceNumber #d3)
47)
48 (DataObject.IsDescribedBy.Attribute R028 ENT02 DMATT15
49 (SequenceNumber #d1)
50)
51 (DataObject.IsDescribedBy.Attribute R029 ENT02 DMATT16
```

```
1 (SequenceNumber #d2)
2)
3 (DataObject.IsDescribedBy.Attribute R030 ENT02 DMATT17
4 (SequenceNumber #d3)
5)
6 (DataObject.IsDescribedBy.Attribute R031 ENT09 DMATT20
7 (SequenceNumber #d1)
8)
9 (Attribute.IsOccurrenceOf.DataType R032 DMATT12 TYPE01)
10 (Attribute.IsOccurrenceOf.DataType R033 DMATT13 TYPE08)
11 (Attribute.IsOccurrenceOf.DataType R034 DMATT14 TYPE09)
12 (Attribute.IsOccurrenceOf.DataType R035 DMATT15 TYPE01)
13 (Attribute.IsOccurrenceOf.DataType R036 DMATT16 TYPE03)
14 (Attribute.IsOccurrenceOf.DataType R037 DMATT17 TYPE04)
15 (Attribute.IsOccurrenceOf.DataType R038 DMATT20 TYPE04)
16)
```

## **A3. Example 2**

### **A3.1 Introduction**

This is an example of a transfer using an extended meta-model.

### **A3.2 Overview**

The extended meta-model adds a new *SubjectArea* called *SecurityExtension*. This *SubjectArea* adds a new *MetaEntity* called *SecurityClassification*, which is related to *SemanticInformationObject* since security of data in a banking information system is paramount, and extends the *MetaEntity* called *Entity* by adding two *MetaAttributes*.

1 **A3.3 SubjectArea Definition**

2 NAME.....**SecurityExtension**  
3 CDIFMETAIDENTIFIER .....NEW00001  
4 DESCRIPTION.....This Subject Area covers the Security Extensions.  
5 USAGE.....  
6 ALIASES .....  
7 CONSTRAINTS.....  
8 VERSIONNUMBER.....01.00

---

---

1 **A3.4 Definition of the Additions**

2 A new *MetaEntity* called *SecurityClassification* is added to the meta-model. It is related to the  
3 *MetaEntity* called *SemanticInformationObject* by a *MetaRelationship* known as  
4 *SemanticInformationObject.IsControlledBy.SecurityClassification*. The definitions of these  
5 additional *MetaEntities* and *MetaRelationships* are shown in the following subsections.

1 **A3.4.1 MetaEntity Definitions**

2 **A3.4.1.1 SecurityClassification**

3 NAME.....**SecurityClassification**  
4 ALSOUSEDIN.....  
5 CDIFMETAI DENTIFIER .....ME001  
6 SUBTYPEOF.....SemanticInformationObject  
7 SUPERTYPEOF .....  
8 DESCRIPTION.....A classification that indicates the security procedures that must be  
9 adhered to when designing systems to process data.  
10 USAGE.....  
11 ALIASES .....SecurityLevel  
12 CONSTRAINTS.....  
13 TYPE .....Kernel  
14

---

15 INHERITED META-ATTRIBUTES

|    |                  |                                |
|----|------------------|--------------------------------|
| 17 | CDIFIdentifier   | from RootObject                |
| 18 | DateCreated      | from RootObject                |
| 19 | DateUpdated      | from RootObject                |
| 20 | TimeCreated      | from RootObject                |
| 21 | TimeUpdated      | from RootObject                |
| 23 | BriefDescription | from SemanticInformationObject |
| 24 | FullDescription  | from SemanticInformationObject |

---

26 LOCAL META-ATTRIBUTES

27 Classification  
28  
29

---

30 INHERITED META-RELATIONSHIPS

31 RootEntity.IsRelatedTo.RootEntity  
32  
33

1 LOCAL META-RELATIONSHIPS  
2       SemanticInformationObject.IsControlledBy.SecurityClassification  
3

---

---

4  
5 META-ATTRIBUTE NAME..... **Classification**  
6 CDIFMETAIDENTIFIER .....MA001  
7 DESCRIPTION.....A name identifying a known procedure to be followed.  
8 USAGE.....  
9 ALIASES .....  
10 CONSTRAINTS.....  
11 DATATYPE .....Enumerated  
12 DOMAIN .....TopSecret, Secret, Classified, Restricted, Unclassified  
13 LENGTH.....  
14 ISOPTIONAL.....False  
15

---

---

1 **A3.4.2 MetaRelationship Definitions**

2 **A3.4.2.1 SemanticInformationObject.IsControlledBy.SecurityClassification**

3 NAME.....IsControlledBy  
4 ALSOUSEDIN.....  
5 CDIFMETAI DENTIFIER .....MR001  
6 SUBTYPEOF.....RootEntity.IsRelatedTo.RootEntity  
7 SUPERTYPEOF.....  
8 MINSOURCECARD.....0  
9 MAXSOURCECARD.....N  
10 MINDESTCARD .....0  
11 MAXDESTCARD .....1  
12 DESCRIPTION.....This relates a SemanticInformationObject to the controlling  
13 SecurityClassification.  
14 USAGE.....  
15 ALIASES.....  
16 CONSTRAINTS.....  
17

---

---

18 INHERITED META-ATTRIBUTES

|    |                |                 |
|----|----------------|-----------------|
| 19 | CDIFIdentifier | from RootObject |
| 20 | DateCreated    | from RootObject |
| 21 | DateUpdated    | from RootObject |
| 22 | TimeCreated    | from RootObject |
| 23 | TimeUpdated    | from RootObject |
| 24 |                |                 |
| 25 |                |                 |

---

26 LOCAL META-ATTRIBUTES

---

---

27  
28  
29  
30

### 1 **A3.5 Definition of the Extensions**

2 The *MetaEntity* called *Entity* is extended by the addition of two *MetaAttributes*:

---

---

3  
4 META-ATTRIBUTE NAME.....**Plural**  
5 CDIFMETAI DENTIFIER .....MA002  
6 DESCRIPTION.....A term to be used when referring to more than one of the subject  
7 entity.  
8 USAGE.....This name will be used when describing the entities in the Data  
9 Model.  
10 ALIASES .....  
11 CONSTRAINTS.....  
12 DATATYPE .....String  
13 DOMAIN .....  
14 LENGTH.....64  
15 ISOPTIONAL.....True  
16

---

17  
18 META-ATTRIBUTE NAME.....**AnnualGrowth**  
19 CDIFMETAI DENTIFIER .....MA003  
20 DESCRIPTION.....The expected increase in the number of occurrences of this entity in  
21 one year.  
22 USAGE.....  
23 ALIASES .....  
24 CONSTRAINTS.....  
25 DATATYPE .....Integer  
26 DOMAIN .....  
27 LENGTH.....  
28 ISOPTIONAL.....True  
29

---

---

## 1 A3.6 The Transfer

```
2 CDIF,SYNTAX "SYNTAX.1" "02.00.00",ENCODING "ENCODING.1" "02.00.00", CODESET
3 "ISO-8859-1"
4 #|Sample CDIF Transfer using extended CDIF Meta-model|#
5 #| Header Section|#
6 (:HEADER
7 (:SUMMARY
8 (ExporterName "CASE Genius")
9 (ExporterVersion "01.00.00")
10 (ExportDate "1991/04/01")
11 (ExportTime "07:00:00")
12 (PublisherName "Mary Lomas")
13)
14)
15
16 #| Meta - model Section|#
17 (:META-MODEL
18 (:SUBJECTAREAREFERENCE Foundation
19 (:VERSIONNUMBER "01.00")
20)
21 (:SUBJECTAREAREFERENCE DataModeling
22 (:VERSIONNUMBER "01.00")
23)
24 (:SUBJECTAREAREFERENCE DataDefinition
25 (:VERSIONNUMBER "01.00")
26)
27 (:SUBJECTAREAREFERENCE Common
28 (:VERSIONNUMBER "01.00")
29)
30)
31 #| The new SubjectArea is defined:|#
32
33 (SubjectArea NEW00001
34 (Name *SecurityExtension*)
35 (VersionNumber "01.00")
36 (Description #[This Subject Area covers the Security
37 Extensions.]#)
38)
39
40 #| Then the new MetaEntity called SecurityClassification is defined:|#
41
42 (MetaEntity ME001
43 (Name *SecurityClassification*)
44 (Description #[A classification that indicates the security
45 procedures that must be adhered to when designing systems to process data.]#)
46 (Aliases "SecurityLevel")
47 (Type <Kernel>)
48)
49
```

```
1 #| SecurityClassification is a subtype of SemanticInformationObject: |#
2 (AttributableMetaObject.HasSubtype.AttributableMetaObject
3 4 ME001)
4
5 (MetaAttribute MA001
6 (Name *Classification*)
7 (Description #[A name identifying a known procedure to be
8 followed.]#)
9 (DataType <Enumerated>)
10 (Domain #[<TopSecret>, <Secret>, <Classified>, <Restricted>,
11 <Unclassified>]#)
12 (IsOptional -False-)
13)
14 (MetaAttribute.IsLocalMetaAttributeOf.AttributableMetaObject
15 MA001 ME001)
16
17 #| Both the meta-entity and its meta-attribute are part of the
18 SecurityExtension SubjectArea: |#
19 (CollectableMetaObject.IsUsedIn.SubjectArea ME001 NEW00001)
20 (CollectableMetaObject.IsUsedIn.SubjectArea MA001 NEW00001)
21
22 #| Then the MetaRelationship called
23 SemanticInformationObject.IsControlledBy.SecurityClassification is defined:
24 |#
25
26 (MetaRelationship MR001
27 (Name *IsControlledBy*)
28 (Description #[This relates a SemanticInformationObject to the
29 controlling SecurityClassification.]#)
30 (MinSourceCard "0")
31 (MaxSourceCard "N")
32 (MinDestCard "0")
33 (MaxDestCard "1")
34)
35
36 #| Its source is SemanticInformationObject: |#
37 (MetaRelationship.HasSource.MetaEntity MR001 4)
38
39 #| Its destination is SecurityClassification: |#
40 (MetaRelationship.HasDestination.MetaEntity MR001 ME001)
41
42 #| It is a subtype of RootEntity.IsRelatedTo.RootEntity: |#
43 (AttributableMetaObject.HasSubtype.AttributableMetaObject
44 3 MR001)
45
46 #| The new MetaRelationship and the MetaEntity called
47 SemanticInformationObject are part of the SecurityExtension SubjectArea: |#
48 (CollectableMetaObject.IsUsedIn.SubjectArea MR001 NEW00001)
49 (CollectableMetaObject.IsUsedIn.SubjectArea 4 NEW00001)
50
```

```
1 #| Finally, the MetaEntity called Entity is extended by adding two
2 MetaAttributes: |#
3
4 (MetaAttribute MA002
5 (Name *Plural*)
6 (Description #[A term to be used when referring to more than one
7 of the subject entity.]#)
8 (Usage #[This name will be used when describing the entities in
9 the Data Model.]#)
10 (DataType <String>)
11 (Length #d64)
12 (IsOptional -True-)
13)
14
15 (MetaAttribute MA003
16 (Name *AnnualGrowth*)
17 (Description #[The expected increase in the number of occurrences
18 of this entity in one year.]#)
19 (DataType <Integer>)
20 (IsOptional -True-)
21)
22
23 #| These meta-attributes are both local meta-attributes of Entity: |#
24 (MetaAttribute.IsLocalMetaAttributeOf.AttributableMetaObject
25 MA002 20)
26 (MetaAttribute.IsLocalMetaAttributeOf.AttributableMetaObject
27 MA003 20)
28
29 #| The two meta-attributes and the meta-entity called Entity are part of the
30 SecurityExtension SubjectArea: |#
31 (CollectableMetaObject.IsUsedIn.SubjectArea MA002 NEW00001)
32 (CollectableMetaObject.IsUsedIn.SubjectArea MA003 NEW00001)
33 (CollectableMetaObject.IsUsedIn.SubjectArea 20 NEW00001)
34)
35
36 #| M o d e l S e c t i o n |#
37 (:MODEL
38 (DataModel MOD01
39 (Name "Example 2")
40 (BriefDescription "The second example Data Model.")
41 (ModelType "Conceptual")
42)
43 (Entity ENT02
44 (Name "Customer Account")
45 (BriefDescription "The bank account of a customer.")
46 (Plural "Customer Accounts")
47 (AnnualGrowth #d4000)
48)
49 (Entity ENT06
50 (Name "Transaction")
51 (BriefDescription "An action generated by an ATM session against a
```

```
1 customer's bank account.")
2 (Plural "Transactions")
3 (AnnualGrowth #d30000)
4)
5 (OrthogonalSubtypeSet OSS01
6 (Name "TransactionType")
7)
8 (Entity ENT07
9 (Name "Card Retention")
10 (BriefDescription "The action generated by retaining the Card
11 within the ATM.")
12 (Plural "Card Retentions")
13)
14 (Entity ENT08
15 (Name "Single Account Transaction")
16 (BriefDescription "A subtype transaction indicating action against
17 one account.")
18 (Plural "Single Account Transactions")
19)
20 (Entity ENT09
21 (Name "Transfer")
22 (BriefDescription "A subtype transaction indicating movement of
23 funds between multiple accounts.")
24 (Plural "Transfers")
25)
26 (LocalAttribute DMATT12
27 (Name "Transaction Identifier")
28 (BriefDescription "The unique identifier of a Transaction.")
29 (IsOptional -False-)
30)
31 (LocalAttribute DMATT13
32 (Name "Transaction Date")
33 (BriefDescription "The date the transaction took place.")
34 (IsOptional -False-)
35)
36 (LocalAttribute DMATT14
37 (Name "Transaction Time")
38 (BriefDescription "The time the transaction took place.")
39 (IsOptional -False-)
40)
41 (LocalAttribute DMATT15
42 (Name "Customer Account Identifier")
43 (BriefDescription "The unique identifier of a Customer Account.")
44 (IsOptional -False-)
45)
46 (LocalAttribute DMATT16
47 (Name "Customer Account Type")
48 (BriefDescription "A code that indicates the type of account.")
49 (IsOptional -False-)
50)
51 (LocalAttribute DMATT17
```

---

```
1 (Name "Customer Account Balance")
2 (BriefDescription "The amount of money in a customer's account.")
3 (IsOptional -False-)
4)
5 (LocalAttribute DMATT20
6 (Name "Amount")
7 (BriefDescription "The amount of money transferred between
8 accounts.")
9 (IsOptional -False-)
10)
11 (Relationship REL04
12 (Name "CarriedOut")
13 (BriefDescription "This relates a Customer Account to the Single
14 Account Transaction performed against it.")
15)
16 (Relationship REL05
17 (Name "Movement")
18 (BriefDescription "This relates a Customer Account to a
19 Transfer.")
20)
21 (Role ROLE09
22 (RoleName "On")
23 (MinOuterCardinality "1")
24 (MaxOuterCardinality "1")
25 (MinInnerCardinality "0")
26 (MaxInnerCardinality "N")
27)
28 (Role ROLE10
29 (RoleName "Is")
30 (MinOuterCardinality "1")
31 (MaxOuterCardinality "1")
32 (MinInnerCardinality "1")
33 (MaxInnerCardinality "1")
34)
35 (Role ROLE11
36 (RoleName "From")
37 (MinOuterCardinality "1")
38 (MaxOuterCardinality "1")
39 (MinInnerCardinality "0")
40 (MaxInnerCardinality "N")
41)
42 (Role ROLE12
43 (RoleName "To")
44 (MinOuterCardinality "1")
45 (MaxOuterCardinality "1")
46 (MinInnerCardinality "0")
47 (MaxInnerCardinality "N")
48)
49 (Role ROLE13
50 (RoleName "Is")
51 (MinOuterCardinality "1")
```

```
1 (MaxOuterCardinality "1")
2 (MinInnerCardinality "1")
3 (MaxInnerCardinality "1")
4)
5 (RolePlayer RP01
6 (Name "Actor")
7)
8 (RolePlayer RP02
9 (Name "Object")
10)
11 (RolePlayer RP03
12 (Name "Actor")
13)
14 (RolePlayer RP04
15 (Name "Sender")
16)
17 (RolePlayer RP05
18 (Name "Receiver")
19)
20 (IntegerType ELEM01
21 (Name "Int")
22 (SignedFlag -True-)
23)
24 (DataElementType TYPE01
25 (Name "Identifier")
26)
27 (DataElementType TYPE03
28 (Name "AccountType")
29)
30 (MoneyType TYPE04
31 (Name "Amount")
32)
33 (DateType TYPE08
34 (Name "Date")
35)
36 (TimeType TYPE09
37 (Name "Time")
38)
39 (DomainGroup DOM57)
40 (DomainValueEnumeration DOM58
41 (Name "Checking")
42)
43 (DomainValueEnumeration DOM59
44 (Name "Savings")
45)
46 (SecurityClassification SEC01
47 (Classification <Classified>)
48)
49 (SecurityClassification SEC02
50 (Classification <Restricted>)
51)
```

---

```
1 (DataModel.IsCollectionOf.DataModelObject R001 MOD01 ENT02)
2 (DataModel.IsCollectionOf.DataModelObject R002 MOD01 ENT06)
3 (DataModel.IsCollectionOf.DataModelObject R003 MOD01 ENT07)
4 (DataModel.IsCollectionOf.DataModelObject R004 MOD01 ENT08)
5 (DataModel.IsCollectionOf.DataModelObject R005 MOD01 ENT09)
6 (DataModel.IsCollectionOf.DataModelObject R006 MOD01 REL04)
7 (DataModel.IsCollectionOf.DataModelObject R007 MOD01 REL05)
8 (InheritableDataModelObject.Has.OrthogonalSubtypeSet R044 ENT06 OSS01)
9 (OrthogonalSubtypeSet.IsConstructedWith.InheritableDataModelObject R008
10 OSS01 ENT07)
11 (OrthogonalSubtypeSet.IsConstructedWith.InheritableDataModelObject R009
12 OSS01 ENT08)
13 (OrthogonalSubtypeSet.IsConstructedWith.InheritableDataModelObject R010
14 OSS01 ENT09)
15 (DataModelObject.Plays.Role R011 ENT02 RP02)
16 (DataModelObject.Plays.Role R012 ENT08 RP01)
17 (DataModelObject.Plays.Role R013 ENT02 RP04)
18 (DataModelObject.Plays.Role R014 ENT02 RP05)
19 (DataModelObject.Plays.Role R015 ENT09 RP03)
20 (RolePlayer.Plays.Role R039 RP01 ROLE10)
21 (RolePlayer.Plays.Role R040 RP02 ROLE09)
22 (RolePlayer.Plays.Role R041 RP03 ROLE13)
23 (RolePlayer.Plays.Role R042 RP04 ROLE11)
24 (RolePlayer.Plays.Role R043 RP05 ROLE12)
25 (Role.BelongsTo.Relationship R016 ROLE09 REL04)
26 (Role.BelongsTo.Relationship R017 ROLE10 REL04)
27 (Role.BelongsTo.Relationship R018 ROLE11 REL05)
28 (Role.BelongsTo.Relationship R019 ROLE12 REL05)
29 (Role.BelongsTo.Relationship R020 ROLE13 REL05)
30 (DataType.HasSubtype.DataType R021 ELEM01 TYPE01)
31 (DataType.TakesValuesFrom.DomainGroup R022 TYPE03 DOM57)
32 (DomainGroup.Contains.Domain R023 DOM57 DOM58)
33 (DomainGroup.Contains.Domain R024 DOM57 DOM59)
34 (DataObject.IsDescribedBy.Attribute R025 ENT06 DMATT12
35 (SequenceNumber #d1)
36)
37 (DataObject.IsDescribedBy.Attribute R026 ENT06 DMATT13
38 (SequenceNumber #d2)
39)
40 (DataObject.IsDescribedBy.Attribute R027 ENT06 DMATT14
41 (SequenceNumber #d3)
42)
43 (DataObject.IsDescribedBy.Attribute R028 ENT02 DMATT15
44 (SequenceNumber #d1)
45)
46 (DataObject.IsDescribedBy.Attribute R029 ENT02 DMATT16
47 (SequenceNumber #d2)
48)
49 (DataObject.IsDescribedBy.Attribute R030 ENT02 DMATT17
50 (SequenceNumber #d3)
```

---

```
1)
2 (DataObject.IsDescribedBy.Attribute R031 ENT09 DMATT20
3 (SequenceNumber #d1)
4)
5 (Attribute.IsOccurrenceOf.DataType R032 DMATT12 TYPE01)
6 (Attribute.IsOccurrenceOf.DataType R033 DMATT13 TYPE08)
7 (Attribute.IsOccurrenceOf.DataType R034 DMATT14 TYPE09)
8 (Attribute.IsOccurrenceOf.DataType R035 DMATT15 TYPE01)
9 (Attribute.IsOccurrenceOf.DataType R036 DMATT16 TYPE03)
10 (Attribute.IsOccurrenceOf.DataType R037 DMATT17 TYPE04)
11 (Attribute.IsOccurrenceOf.DataType R038 DMATT20 TYPE04)
12 (SemanticInformationObject.IsControlledBy.SecurityClassification R039
13 ENT02 SEC02)
14 (SemanticInformationObject.IsControlledBy.SecurityClassification R040
15 ENT06 SEC01)
16 (SemanticInformationObject.IsControlledBy.SecurityClassification R041
17 ENT07 SEC02)
18 (SemanticInformationObject.IsControlledBy.SecurityClassification R042
19 ENT08 SEC02)
20 (SemanticInformationObject.IsControlledBy.SecurityClassification R043
21 ENT09 SEC02)
22 (SemanticInformationObject.IsControlledBy.SecurityClassification R044
23 DMATT17 SEC01)
24 (SemanticInformationObject.IsControlledBy.SecurityClassification R045
25 DMATT20 SEC01)
26)
```

1

This page is intentionally blank



1  
2  
**APPENDIX B<sup>1</sup>**  
**QUESTIONS AND ANSWERS**

3 The following questions and answers are presented in order to help the reader's understanding of  
4 the standard. They have been derived from discussions held within the CDIF Technical  
5 Committee and presentations to outside observers.

6 **Question 1:**

7 **What is the advantage of a clear text encoding?**

8 **Answer:**

9 Different encodings address different requirements. For example:

10 A **Clear Text** encoding is human readable but very bulky.

11 A **Character** encoding (of which Clear Text is a special case) is also human readable,  
12 more compact than Clear Text, and can pass unhindered down communications lines.  
13 However, it is more terse than Clear Text and requires detailed knowledge of the encoding  
14 to decipher.

15 A **Binary** encoding is more compact than Character, but is not human readable, and may  
16 not pass down communications lines unhindered.

17 An **Encrypted** encoding may be based on any of the three previous classifications, but is  
18 defined with an encryption algorithm for security purposes.

19 **Question 2:**

20 **Is the encoding described in this document the only CDIF encoding?**

21 **Answer:**

22 Initially, the CDIF Technical Committee is providing only one standard CDIF encoding.  
23 However, further CDIF encodings may be made available in the future.

---

<sup>1</sup>This appendix is not a formal part of the attached EIA standard, but is included for purposes of information only.

1 **Question 3:**

2 **Are non-standard encodings allowed?**

3 **Answer:**

4 Non-standard encodings may be used by mutual agreement between exporting and importing  
5 tools. However, it should be stressed that the use of a non-standard encoding will severely  
6 restrict the useability of any transfer. A standard CDIF encoding should be used wherever  
7 possible.

8 **Question 4:**

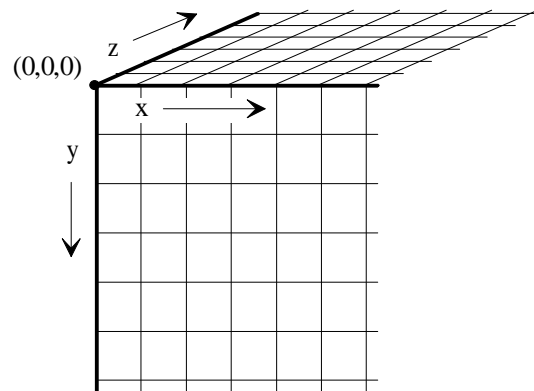
9 **How do I encode text that has non-printable characters?**

10 **Answer:**

11 These characters are encoded using a sequence of characters. Let's say the character you want to  
12 encode is a control-r. Control-r is normally represented in binary as 00010010, which is the  
13 decimal 18. And, since the decimal number 18 is 12 in hexadecimal, the control-r would be  
14 encoded as the character sequence \#H12\.

## GLOSSARY

|                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ATTRIBUTE</b>                | A single-valued characteristic of an entity or relationship.                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>CDIF</b>                     | See <i>CDIF Family of Standards</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>CDIF CLEAR TEXT ENCODING</b> | A human readable encoding for the CDIF syntax, SYNTAX.1.                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>CDIF COORDINATE FRAME</b>    | <p>The system of graphics addressing used by CDIF. Graphics space is defined as a coordinate system that is a three dimensional rectilinear grid. The intersections of grid lines define addresses (CDIF Points). The intersection of any two grid lines defines a coordinate plane. Each grid cell in a coordinate plane is a CDIF Pixel. All grid coordinates are integers and all grid lines are infinitely thin.</p> <p>The figure below describes the CDIF Coordinate Frame.</p> |



---

<sup>1</sup>This glossary is not a formal part of the attached EIA standard, but is included for purposes of information only.

**CDIF FAMILY OF STANDARDS**

The CDIF Family of Standards is composed of a set of standards that, when used together, provide a standard definition for the interchange of information between CASE tools.

**CDIF IDENTIFIER**

An attribute that uniquely identifies an object in the Model Section of a transfer.

**CDIF INTEGRATED META-MODEL**

The description of the set of concepts and notations used to define a model. The CDIF Integrated Meta-model defines an Entity-Relationship-Attribute model that is used to construct and define models used in systems development.

**CDIF METAIDENTIFIER**

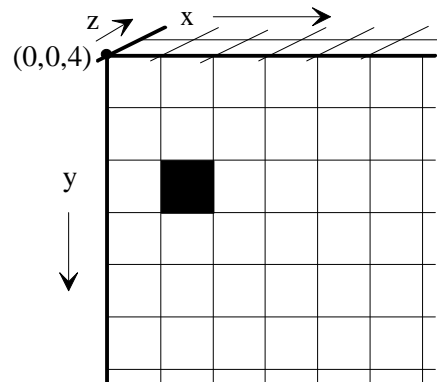
A meta-meta-attribute that uniquely identifies a meta-object in the Meta-model Section of a transfer.

**CDIF META-META-MODEL**

The description of the set of concepts and notations used to define a meta-model. Specifically, the CDIF Meta-meta-model defines an Entity-Relationship-Attribute model that is used to construct and define both meta-models and the CDIF Meta-meta-model itself.

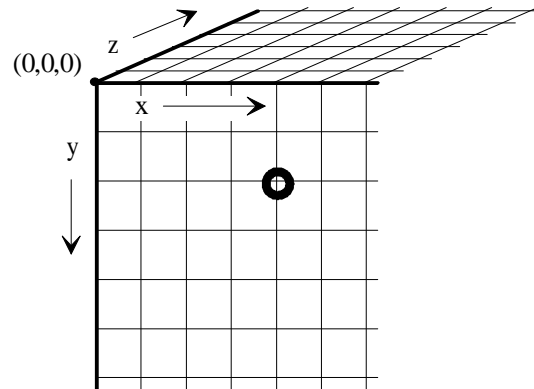
**CDIF PIXEL**

A grid cell on the coordinate plane in the CDIF Coordinate Frame. The shaded area in the figure below is a CDIF Pixel in the  $(x,y,4)$  coordinate plane.



**CDIF POINT**

The location of a point in 3-dimensional space as specified by the  $x,y,z$  coordinates. The circle in the figure below encloses a point located at  $(4,2,0)$ .



**CDIF TRANSFER**

A combination of a particular syntax, a particular encoding of that syntax, and a meta-model. In other words, a complete definition of the format and contents of a transfer.

|                                          |                                                                                                                                                                                                                               |
|------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>CDIF TRANSFER FORMAT</b>              | A combination of a particular syntax and a particular encoding of that syntax which together provides a complete definition of the transfer format.                                                                           |
| <b>CDIF TRANSFER SYNTAX AND ENCODING</b> | A standard vehicle format supported by CDIF. The combination of SYNTAX.1 and ENCODING.1 forms the initial CDIF Transfer Syntax and Encoding.                                                                                  |
| <b>CHARACTER SET</b>                     | A character set is a collection of characters used in an Encoding to represent terminal symbols. The character set used is significant in the encoding of text and string meta-attributes for a CDIF Transfer.                |
| <b>CLEAR TEXT</b>                        | A form of encoding where the resulting physical file is human-readable.                                                                                                                                                       |
| <b>COORDINATE PLANE</b>                  | See <i>CDIF Coordinate Frame</i> .                                                                                                                                                                                            |
| <b>ENCODING</b>                          | An encoding defines how the elements of a syntax are physically represented using an identified character set. Details of representation of the various terminal symbols and data types in the syntax's grammar are provided. |
| <b>ENCODING.1</b>                        | The CDIF Family of Standards supports multiple transfer formats, each composed of a syntax and an encoding. ENCODING.1 is the primary encoding defined within the CDIF Family of Standards.                                   |
| <b>ENTITY</b>                            | An object (i.e., thing, event or concept) that occurs in a model (i.e., transfer).                                                                                                                                            |
| <b>INFORMATION CONTENT</b>               | The Information Content is the set of meta-model and model instances found in a CDIF Transfer.                                                                                                                                |

**INSTANCE**

An individual occurrence of a type (e.g., the diagram *MyDiagram* is an instance of the *Diagram* type).

Usually, an instance is an object which occurs one meta-level beneath its (type's) definition. This is not always true, however (e.g., within the CDIF presentation subject areas where icon type definitions and their instances are both contained within the same meta-level).

**INTEGRATED META-MODEL**

See *CDIF Integrated Meta-model* .

**META**

According to Webster's Ninth New Collegiate Dictionary: *more comprehensive: transcending - used with the name of a discipline to designate a new but related discipline designed to deal critically with the original one.*

Meta- is used in CDIF generally as a prefix to a concept to imply definition information about the concept. Specifically, used to designate the location of an object in the three model layers.

**META-ATTRIBUTE**

A definition of a characteristic of a meta-entity or meta-relationship. Instances of a meta-attribute occur in a model as data values.

**META-ENTITY**

A definition of a type of data object that occurs in CDIF models. Specifically, a meta-entity represents a set of zero or more meta-attributes, stored together to represent a thing, event or concept that has instances in a model.

**META-LAYERS**

See *Model Layers*.

|                               |                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>META-META-ATTRIBUTE</b>    | A definition of a characteristic of a meta-meta-entity or meta-meta-relationship. Instances of a meta-meta-attribute occur in a meta-model as meta-data values.                                                                                                                                                                                                                                                                       |
| <b>META-META-ENTITY</b>       | A definition of the behavior and structure of meta-entities, meta-relationships, meta-attributes, or subject areas (i.e., a definition of the meta-object definitions used to describe information in models).                                                                                                                                                                                                                        |
| <b>META-META-MODEL</b>        | See <i>CDIF Meta-meta-model</i> .                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>META-META-RELATIONSHIP</b> | A definition of a type of data object that occurs in CDIF meta-models. Specifically, a meta-meta-relationship represents the definition of a relationship between instances of meta-meta-entities.                                                                                                                                                                                                                                    |
| <b>META-MODEL</b>             | <p>A meta-model contains detailed definitions of the meta-entities, meta-relationships and meta-attributes whose instances represent an actual CDIF transfer.</p> <p>The CDIF Integrated Meta-model (as defined in the set of standards that comprise the CDIF Family of Standards) is a definition of all of the types of information that can be transferred in a CDIF Transfer without using the CDIF extensibility mechanism.</p> |
| <b>META-OBJECT</b>            | A meta-object is a generic term for meta-entities and meta-relationships                                                                                                                                                                                                                                                                                                                                                              |
| <b>META-RELATIONSHIP</b>      | A definition of a type of data object that occurs in CDIF models. Specifically, a meta-relationship represents the definition of a relationship between meta-entities that has instances in a model. A meta-relationship may also define a set of zero or more meta-attributes, stored together to represent characteristics of a relationship between meta-entities.                                                                 |

**MODEL**

A specific collection of software engineering data. This is called a model because it usually represents a model of a software system under development.

A collection of instances of meta-objects.

**MODEL LAYERS**

The different layers of definition (or abstraction) used in defining the CDIF Family of Standards. The four model layers in CDIF are: user data, model, meta-model, meta-meta-model.

Any given model layer provides an accurate and complete definition of all the instances that may occur one layer below the given layer. For example, the meta-meta-model provides a set of definitions that are used to construct and understand the meta-model; the meta-model provides a set of definitions that are used to construct and understand a model.

**NON-TERMINAL SYMBOL**

A part of the hierarchical definition of a syntax that is further decomposed in the hierarchy.

**PRODUCTION RULE**

The definition of a syntactic element in the form of an expression consisting of characters, character strings, and other syntactic elements.

**RELATIONSHIP**

A real-world association among one or more entities. Where the association is between an entity and itself, the relationship is said to be recursive.

|                           |                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SUBJECT AREA</b>       | A related collection of meta-object instance definitions. Subject areas are used to define scoped areas of interest. Subject areas overlap to ensure the integration of the overall Meta-model, but a tool need only use those subject areas relevant to the data to be exported/imported.                                                                                                 |
| <b>SYNTAX</b>             | A syntax is a definition of the format of information in a CDIF transfer. The definition is in the form of a grammar. It is specified with regard to ordering and repetition, down to the level of terminals, but does not specify the representation of any of the terminal objects in the grammar. Those decisions are defined in the encoding of the syntax.                            |
| <b>SYNTAX.1</b>           | The CDIF Family of Standards supports multiple transfer formats, each composed of a syntax and an encoding. SYNTAX.1 is the primary syntax defined within the CDIF Family of Standards.                                                                                                                                                                                                    |
| <b>TERMINAL SYMBOL</b>    | A part of the hierarchical definition of a syntax that is <b>not</b> further decomposed in the hierarchy.                                                                                                                                                                                                                                                                                  |
| <b>TRANSFER</b>           | See <i>CDIF Transfer</i> .                                                                                                                                                                                                                                                                                                                                                                 |
| <b>TRANSFER FORMAT</b>    | See <i>CDIF Transfer Format</i> .                                                                                                                                                                                                                                                                                                                                                          |
| <b>WORKING META-MODEL</b> | The working meta-model is the definition of the specific meta-objects that may be instantiated in the model section of a CDIF Transfer. The working meta-model comprises the meta-objects in the CDIF Integrated Meta-model that are used by the subject areas referenced in the meta-model section of the transfer, and the meta-objects defined as extensions in the meta-model section. |

1

This page is intentionally blank



## INDEX

### Special Characters

- !! 6
- ... 6
- ::= 6
- <> 5
- [] 6
- { } 6
- | 6

### A

- Alternative operator 6
- Angle braces 5
- ATTRIBUTE**
  - GLOSSARY ENTRY 55

### B

- Backus Naur Form (See BNF)
- Binary encoding 53
- BNF
  - (See also Formal grammar)
  - conventions 5–6
- Bold type 5, 6
- Braces
  - angle 5
  - curly 6
  - square 6

### C

- Carriage return 7, 28
- Case sensitivity 20, 21
- CDIF**
  - GLOSSARY ENTRY 55
- CDIF CLEAR TEXT ENCODING**
  - GLOSSARY ENTRY 55
- CDIF COORDINATE FRAME**
  - GLOSSARY ENTRY 56
- CDIF Family of Standards 1
  - GLOSSARY ENTRY 56
- CDIF IDENTIFIER**

- GLOSSARY ENTRY 56
- CDIF INTEGRATED META-MODEL**
  - GLOSSARY ENTRY 56
- CDIF METAIDENTIFIER**
  - GLOSSARY ENTRY 57
- CDIF META-META-MODEL**
  - GLOSSARY ENTRY 57
- CDIF POINT**
  - GLOSSARY ENTRY 58
- CDIF TRANSFER**
  - GLOSSARY ENTRY 58
- CDIF TRANSFER FORMAT**
  - GLOSSARY ENTRY 58
- CDIF TRANSFER SYNTAX AND ENCODING**
  - GLOSSARY ENTRY 58
- Character
  - carriage return 7, 28
  - form feed 7, 28
  - line feed 7, 28
  - tab
    - horizontal 7, 28
    - vertical 7, 28
- Character encoding 53
- Character Set
  - glossary entry 58
- Clear Text 9, 53
  - GLOSSARY ENTRY 58
  - objectives 7
- Comment operator 6
- Conventions 5
  - BNF 5–6
- COORDINATE PLANE**
  - GLOSSARY ENTRY 59
- Curly braces 6

### D

- Definition operator 6
- Document
  - structure 3–4

### E

- Element
  - syntactic 5
- Ellipsis 6

---

|    |                                     |    |                                   |
|----|-------------------------------------|----|-----------------------------------|
| 1  | Encoding                            | 49 | Exp 11, 23                        |
| 2  | binary 53                           | 50 | Exponent 11, 23                   |
| 3  | character 53                        | 51 | ExtendMetaAttributeKeyword 20, 23 |
| 4  | clear text 53                       | 52 | FalseValue 23                     |
| 5  | encrypted 53                        | 53 | FloatDesignator 11, 23            |
| 6  | <b>GLOSSARY ENTRY</b> 59            | 54 | FloatValue 11, 23                 |
| 7  | identifier 7                        | 55 | GeneralPrintableChar 23           |
| 8  | non-standard 54                     | 56 | HeaderKeyword 20, 23              |
| 9  | standard 54                         | 57 | HeightKeyword 20, 23              |
| 10 | <b>ENCODING.1</b>                   | 58 | HexadecimalValue 23               |
| 11 | <b>GLOSSARY ENTRY</b> 59            | 59 | HexDecimalValue 10                |
| 12 | Encrypted encoding 53               | 60 | HexDigit 10, 23                   |
| 13 | <b>ENTITY</b>                       | 61 | HexNumber 10, 23                  |
| 14 | <b>GLOSSARY ENTRY</b> 59            | 62 | HexRadix 10, 23                   |
| 15 | Expression 5                        | 63 | Hours 12, 24                      |
| 16 | <b>F</b>                            | 64 | Hyphen 24                         |
| 17 | Form feed 7, 28                     | 65 | Identifier 19, 24                 |
| 18 | Formal grammar 21–28                | 66 | IdentifierValue 13, 24            |
| 19 | (See also BNF)                      | 67 | Integer 9, 24                     |
| 20 | Absolute 21                         | 68 | IntegerListKeyword 20, 24         |
| 21 | AbsoluteLocal 21                    | 69 | ListSeparator 24                  |
| 22 | AbsoluteUTC 21                      | 70 | LowerCaseAlphabeticChar 24        |
| 23 | BinaryDigit 9, 21                   | 71 | Mantissa 11, 24                   |
| 24 | BinaryNumber 9, 21                  | 72 | MetaMetaObjectName 18, 24         |
| 25 | BinaryRadix 9, 21                   | 73 | MetaModelKeyword 20, 24           |
| 26 | BinaryValue 9, 21                   | 74 | MetaObjectName 18, 24             |
| 27 | BitmapKeyword 20, 21                | 75 | Minutes 12, 24                    |
| 28 | CloseComment 17, 21                 | 76 | ModelKeyword 20, 24               |
| 29 | CloseScope 21                       | 77 | MonthValue 12, 24                 |
| 30 | CloseText 15, 21                    | 78 | MultibyteIdentifier 20, 25        |
| 31 | Comment 16, 21                      | 79 | OctalDigit 10, 25                 |
| 32 | CommentBody 16, 21                  | 80 | OctalNumber 10, 25                |
| 33 | CommentBodyChar 17, 22              | 81 | OctalRadix 10, 25                 |
| 34 | CommentBodyCharExcludingHashSign 22 | 82 | OctalValue 10, 25                 |
| 35 | Date 12, 22                         | 83 | OpenComment 16, 25                |
| 36 | DateDelimiter 12, 22                | 84 | OpenScope 25                      |
| 37 | DateKeyword 20, 22                  | 85 | OpenText 14, 25                   |
| 38 | DayValue 12, 22                     | 86 | OtherChar 15, 25                  |
| 39 | DecimalIntegerValue 9               | 87 | PixelIntensity 25                 |
| 40 | DecimalPoint 11, 22                 | 88 | PixelSeparator 25                 |
| 41 | DecimalRadix 9, 22                  | 89 | PointKeyword 20, 25               |
| 42 | Digit 22                            | 90 | PointListKeyword 20, 25           |
| 43 | Dot 22                              | 91 | PointSeparator 25                 |
| 44 | EncodingID 7                        | 92 | PositiveInteger 25                |
| 45 | EncodingVersion 7                   | 93 | RelativeNegative 25               |
| 46 | EnumeratedSeparator 22              | 94 | RelativePositive 25               |
| 47 | EnumeratedValue 13, 23              | 95 | Seconds 12, 26                    |
| 48 | EscapeCharacter 23                  | 96 | Sign 26                           |
|    |                                     | 97 | Space 26                          |
|    |                                     | 98 | String 14, 26                     |
|    |                                     | 99 | StringDelimiter 14, 26            |

---

1 StringElement 14, 26  
2 SubjectAreaReferenceKeyword 20, 26  
3 SummaryKeyword 20, 26  
4 TextElement 15, 26  
5 TextElementExcludingHashSign 15, 27  
6 TextString 14, 27  
7 Time 12, 27  
8 TimeDelimiter 12, 27  
9 TimeKeyword 20, 27  
10 TrueValue 27  
11 UnderScore 27  
12 UpperCaseAlphabeticChar 27  
13 UpperOrLowerCaseAlphabeticChar 27  
14 VersionNumberKeyword 20, 27  
15 WhiteSpace 27  
16 WidthKeyword 20, 28  
17 YearValue 12, 28  
18 Formal grammr  
19 CommentBodyCharExcludingHashSign 17

## 20 G

21 Grammar 5

## 22 H

## 23 I

24 **INFORMATION CONTENT**  
25 **GLOSSARY ENTRY 59**  
26 **INSTANCE**  
27 **GLOSSARY ENTRY 59**  
28 **INTEGRATED META-MODEL**  
29 **GLOSSARY ENTRY 59**

## 30 J

## 31 K

32 Keyword 19–20

## 33 L

34 Line feed 7, 28

## 35 M

36 **META-**  
37 **GLOSSARY ENTRY 60**  
38 **META-ATTRIBUTE**  
39 **GLOSSARY ENTRY 60**  
40 **META-ENTITY**  
41 **GLOSSARY ENTRY 60**  
42 Meta-language 5  
43 **META-LAYERS**  
44 **GLOSSARY ENTRY 60**  
45 **META-META-ATTRIBUTE**  
46 **GLOSSARY ENTRY 60**  
47 **META-META-ENTITY**  
48 **GLOSSARY ENTRY 60**  
49 **META-META-MODEL**  
50 **GLOSSARY ENTRY 60**  
51 Meta-meta-object  
52 name 5  
53 **META-META-RELATIONSHIP**  
54 **GLOSSARY ENTRY 61**  
55 Meta-model  
56 extended 36  
57 **GLOSSARY ENTRY 61**  
58 Meta-object  
59 **GLOSSARY ENTRY 61**  
60 name 5  
61 **META-RELATIONSHIP**  
62 **GLOSSARY ENTRY 61**  
63 **MODEL**  
64 **GLOSSARY ENTRY 61**  
65 **MODEL LAYERS**  
66 **GLOSSARY ENTRY 62**

## 67 N

68 Name  
69 meta-meta-object 5  
70 meta-object 5  
71 **NON-TERMINAL SYMBOL**  
72 **GLOSSARY ENTRY 62**  
73 Notational conventions 5

## 74 O

|    |                           |    |                                 |
|----|---------------------------|----|---------------------------------|
| 1  | <b>P</b>                  | 36 | <b>U</b>                        |
| 2  | Prerequisites 3           |    |                                 |
| 3  | Production rule 5         | 37 | <b>V</b>                        |
| 4  | <b>GLOSSARY ENTRY</b> 62  |    |                                 |
| 5  | <b>Q</b>                  | 38 | Vertical bar 6                  |
| 6  | Quotes 5                  | 39 | <b>W</b>                        |
| 7  | <b>R</b>                  | 40 | Whitespace 7, 9, 14             |
| 8  | Range 6                   | 41 | carriage return 7, 28           |
| 9  | <b>RELATIONSHIP</b>       | 42 | form feed 7, 28                 |
| 10 | <b>GLOSSARY ENTRY</b> 62  | 43 | in example 30                   |
| 11 | Rule                      | 44 | line feed 7, 28                 |
| 12 | production 5              | 45 | non-terminal 15, 17, 22, 26, 27 |
| 13 | <b>S</b>                  | 46 | production 27                   |
| 14 | Square braces 6           | 47 | tab                             |
| 15 | <b>SUBJECT AREA</b>       | 48 | horizontal 7, 28                |
| 16 | <b>GLOSSARY ENTRY</b> 62  | 49 | vertical 7, 28                  |
| 17 | Syntactic element 5       | 50 | <b>WORKING META-MODEL</b>       |
| 18 | <b>SYNTAX</b>             | 51 | <b>GLOSSARY ENTRY</b> 63        |
| 19 | <b>GLOSSARY ENTRY</b> 63  | 52 | <b>X</b>                        |
| 20 | <b>SYNTAX.1</b>           | 53 | <b>Y</b>                        |
| 21 | <b>GLOSSARY ENTRY</b> 63  | 54 | <b>Z</b>                        |
| 22 | <b>T</b>                  | 55 |                                 |
| 23 | Tab                       |    |                                 |
| 24 | horizontal 7, 28          |    |                                 |
| 25 | vertical 7, 28            |    |                                 |
| 26 | Terminal symbol 6         |    |                                 |
| 27 | <b>GLOSSARY ENTRY</b> 63  |    |                                 |
| 28 | Token 9                   |    |                                 |
| 29 | separation of 7, 9        |    |                                 |
| 30 | Transfer 7, 9, 30, 36, 54 |    |                                 |
| 31 | examples 30–51            |    |                                 |
| 32 | file 7                    |    |                                 |
| 33 | <b>GLOSSARY ENTRY</b> 63  |    |                                 |
| 34 | Transfer Format i         |    |                                 |
| 35 | <b>GLOSSARY ENTRY</b> 63  |    |                                 |



This page is intentionally blank

## **RELATED EIA STANDARDS**

1

*EIA/IS-106 CDIF - CASE Data Interchange Format - Overview*

2

*EIA/IS-107 CDIF - Framework for Modeling and Extensibility*

3

*EIA/IS-108 CDIF - Transfer Format - General Rules for Syntaxes and Encodings*

4

*EIA/IS-109 CDIF - Transfer Format - Syntax SYNTAX.1*

5