Final Committee Draft **ISO/IEC FCD 14957**	
Date: **2007-12-23**	Reference number: ISO/JTC 1/SC 32**N1678**
Supersedes document SC 32N1399	

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

ISO/IEC JTC 1/SC 32 Data Management and Interchange Secretariat: USA (ANSI)	Circulated to P- and O-members, and to technical committees and organizations in liaison for voting (P-members only) by: **2008-04-23** Please return all votes and comments in electronic form directly to the SC 32 Secretariat by the due date indicated.

ISO/IEC: FCD 14957:2008(E)

Title: Information technology - Representation of data elements values: Notation of the format 2nd Edition

Project: 1.32.19.02.00.00

Introductory note: The disposition of comments on the CD is provided in 32N1679. The attached document is hereby submitted for a four-month letter ballot to the National Bodies of ISO/IEC JTC 1/SC 32. The ballot starts 2007-12-23.

Medium: E

No. of pages: 13

Dr. Timothy Schoechle, Secretary, ISO/IEC JTC 1/SC 32
Farance Inc *, 3066 Sixth Street, Boulder, CO, United States of America
Telephone: +1 303-443-5490; E-mail: Timothy@Schoechle.org
available from the JTC 1/SC 32 WebSite http://www.jtc1sc32.org/
*Farance Inc. administers the ISO/IEC JTC 1/SC 32 Secretariat on behalf of ANSI

© ISO 2007 – All rights reserved

Reference number of working document: **ISO/IEC JTC1 SC32 N1678**

Date: 2007-12-23

Reference number of document: **ISO/IEC FCD 14957**

Committee identification: ISO/IEC JTC1 SC32 WG2

SC32 Secretariat: US

Information technology — Representation of data elements values: Notation of the format - 2nd Edition

Warning

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: International standard
Document subtype: if applicable
Document stage: (40) Final Committee
Document language: E

Copyright notice

This ISO document is a working draft or committee draft and is copyright-protected by ISO. While the reproduction of working drafts or committee drafts in any form for use by participants in the ISO standards development process is permitted without prior permission from ISO, neither this document nor any extract from it may be reproduced, stored or transmitted in any form for any other purpose without prior written permission from ISO.

Requests for permission to reproduce this document for the purpose of selling it should be addressed as shown below or to ISO's member body in the country of the requester:

ISO copyright office
Case postale 56
CH-1211 Geneva 20
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Reproduction for sales purposes may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

© ISO 2007 – All rights reserved

Contents

Page

© ISO 2007 – All rights reserved

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 14957 was prepared by Technical Committee ISO/IEC JTC1, *Information Technology*, Subcommittee SC32, *Data Management and Interchange*.

© ISO 2007 – All rights reserved

Introduction

Data interchange is experiencing rapid expansion, in the commercial, technical and public sectors. It gives rise to inter working between different communities which often have developed independently information processing applications and telecommunication networks which meet specific needs. Hence an overall situation which suffers from a lack of homogeneity.

In order to remedy this situation, an urgent standardization effort focused in particular on the representation of data elements is necessary.

The representation of a data element supposes in the first place that the format, i.e. the type of characters used in the representation and in the length of the latter is specified. So that these specifications have the same significance for everyone involved, it is necessary to express them in accordance with standardized conventions.

Such rules are likely to eliminate any and all risk of ambiguity, lack of understanding and error; they also facilitate the comparison of data element dictionaries, the design and creation of information systems as well as electronic data interchange (EDI).

These notations have been partially and variously expressed in different International Standards according to the specific contexts in which they have been defined, e.g. EDIFACT (ISO 9735), Banking Standards (as ISO 7982-1), Character sets (ISO 8859), Information processing (ISO 6093), programming languages (ISO/IEC 9899).

Therefore, the objective of this International Standard is to provide a unique source of reference on this issue for all Standards utilizing these type of notations independently of their environments.

© ISO 2007 – All rights reserved

Information technology — Representation of data elements values: Notation of the format - 2nd Edition

1 Scope

This International Standard specifies the notation to be used for stating the format, i.e. the character classes used in the representation of data elements and the length of these representations. It also specifies additional notations relative to the representation of numerical figures.

The scope of this International Standard is limited to graphic characters, such as digits, letters and special characters. It does not specify usage of control characters.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 2382-04, *Information technology — Vocabulary — Part 04: Organization of data*

ISO 6093:1985, *Information processing — Representation of numerical values in character strings for information interchange*

ISO/IEC 11179-3:2003, *Information technology — Metadata Registries (MDR) — Part 3: Registry metamodel and basic attributes*

ISO/IEC 19773-001, *Information technology — Metadata Interoperability and Bindings (MDIB) — Overview*

3 Terms and definitions

For the purposes of this document, the following terms, abbreviations, and definitions apply.

3.1
data element
unit of data that is considered, in context, to be indivisible [ISO/IEC 2382-04:1999]

3.2
character set
finite set of characters that is complete for a given purpose [ISO/IEC 2382-04:1999]

3.3
character type
set of characters of the same kind or having the same use

© ISO 2007 – All rights reserved

EXAMPLE Letters, figures, special characters, etc.

3.4
length (of representation)
number of characters used to represent a data element

4 Notation relative to character types and length of representation of a data element

The format shall be a characterstring sequence. The format is composed of zero or more directives: one or more white-space characters, an ordinary character (neither `%` nor a white-space character), or a conversion specification. Each conversion specification is introduced by the character `%`.

NOTE This specification of formats is based upon the "fscanf()" function in the C programming language (ISO/IEC 9899:1999).

After the %, the following appear in sequence:

— An optional assignment-suppressing character `*`.

— An optional nonzero decimal integer that specifies the maximum field width (in characters).

— An optional length modifier that specifies the size of the receiving object.

— A conversion specifier character that specifies the type of conversion to be applied.

Each directive of the format is processed in turn.

A directive composed of white-space character(s) is executed by reading input up to the first non-white-space character (which remains unread), or until no more characters can be read.

A directive that is an ordinary character is executed by reading the next characters of the stream. If any of those characters differ from the ones composing the directive, the directive fails and the differing and subsequent characters remain unread.

Similarly, if end-of-file, an encoding error, or a read error prevents a character from being read, the directive fails.

A directive that is a conversion specification defines a set of matching input sequences, as described below for each specifier. A conversion specification is executed in the following steps:

Input white-space characters are skipped, unless the specification includes a `[`, `c`, or `n` specifier.[1]

An input item is read from the stream, unless the specification includes an `n` specifier. An input item is defined as the longest sequence of input characters which does not exceed any specified field width and which is, or is a prefix of, a matching input sequence.

[1] These white-space characters are not counted against a specified field width.

© ISO 2007 – All rights reserved

The first character, if any, after the input item remains unread. If the length of the input item is zero, the execution of the directive fails; this condition is a matching failure unless end-of-file, an encoding error, or a read error prevented input from the stream, in which case it is an input failure.

Except in the case of a `%` specifier, the input item (or, in the case of a `%n` directive, the count of input characters) is converted to a type appropriate to the conversion specifier. If the input item is not a matching sequence, the execution of the directive fails: this condition is a matching failure. Unless assignment suppression was indicated by a `*`, the result of the conversion is placed in the object pointed to by the first argument following the format argument that has not already received a conversion result. If this object does not have an appropriate type, or if the result of the conversion cannot be represented in the object, the behavior is undefined.

The conversion specifiers and their meanings are:

— `d` Matches an optionally signed decimal integer, whose format is the same as expected for the subject sequence of the strtol function with the value 10 for the base argument. The corresponding argument shall be a pointer to signed integer.

— `i` Matches an optionally signed integer, whose format is the same as expected for the subject sequence of the strtol function with the value 0 for the base argument. The corresponding argument shall be a pointer to signed integer.

— `o` Matches an optionally signed octal integer, whose format is the same as expected for the subject sequence of the strtoul function with the value 8 for the base argument. The corresponding argument shall be a pointer to unsigned integer.

— `u` Matches an optionally signed decimal integer, whose format is the same as expected for the subject sequence of the strtoul function with the value 10 for the base argument. The corresponding argument shall be a pointer to unsigned integer.

— `x` Matches an optionally signed hexadecimal integer, whose format is the same as expected for the subject sequence of the strtoul function with the value 16 for the base argument. The corresponding argument shall be a pointer to unsigned integer.

— `a,e,f,g` Matches an optionally signed floating-point number, infinity, or NaN, whose format is the same as expected for the subject sequence of the strtod function. The corresponding argument shall be a pointer to floating.

— `c` Matches a sequence of characters of exactly the number specified by the field width (1 if no field width is present in the directive). If an l length modifier is present, the input shall be a sequence of characters.

— `s` Matches a sequence of non-white-space characters. If an l length modifier is present, the input shall be a sequence of characters.

— `[` Matches a nonempty sequence of characters from a set of expected characters (the scanset). The conversion specifier includes all subsequent characters in the format string, up to and including the matching right bracket (`]`). The characters between the brackets (the scanlist) compose the scanset, unless the character after the left bracket is a circumflex (^), in which case the scanset contains all characters that do not appear in the scanlist between the circumflex and the right bracket. If the conversion specifier begins with [] or [^], the right bracket character is in the scanlist and the next following right bracket character is the matching right bracket that ends the specification; otherwise the first following right bracket character is the one that ends the specification. If a - character is in the

© ISO 2007 – All rights reserved

scanlist and is not the first, nor the second where the first character is a `^`, nor the last character, the behavior is implementation-defined.

— `p` Matches an implementation-defined set of sequences that represent a pointer. The input item is converted to a pointer value in an implementation-defined manner.

— `n` No input is consumed.

— `%` Matches a single `%` character; no conversion occurs. The complete conversion specification shall be `%%`.

5 Examples

The following are examples of the syntax.

EXAMPLE 1 The string `"%d"` corresponds to a decimal integer value such as `17`, `0`, `-17`, `017`, but not `1.0`, `ABC`.

EXAMPLE 2 The string `"%03d"` corresponds to a decimal integer value such as `017`, `000`, `017`, but not `1000`.

EXAMPLE 3 The string `"%s"` corresponds to a string value such as `"abcd"`, `"123"`, and `""`.

EXAMPLE 4 The string `"%f"` corresponds to a decimal real value such as `1.0`, `1.`, `-1`.

© ISO 2007 – All rights reserved

Annex A (informative) EBNF grammar for data element description

The following syntax is based upon ISO/IEC 14977, *Information technology — Syntactic metalanguage — Extended BNF*.

```
format_description = { [non_format_component] , format_component } ,
                     [non_format_component];

non_format_component = { non_format_characters };

non_format_characters = ( ascii - "%" ) | "%%" ;

format_component = signed_decimal_format |
                   signed_integer_format |
                   signed_octal_format |
                   unsigned_integer_format |
                   signed_hexadecimal_format |
                   character_array_format |
                   character_string_format |
                   defined_character_set_format |
                   real_number_format |
                   object_pointer_format |
                   null_conversion ;

decimal_with_optinal_leading_zero = decimal_numeral , { decimal_numeral } ;

decimal_numeral = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" ;

signed_decimal_format = "%" , [ "+" | "-" ] ,
                     [  "*" , decimal_with_optional_leading_zero ] , "d" ;

signed_integer_format = "%" , [ "+" | "-" ] ,
                     [  "*" , decimal_with_optional_leading_zero ] , "i" ;

signed_octal_format = "%" , [ "+" | "-" ] ,
                     [  "*" , decimal_with_optional_leading_zero ] , "o" ;

unsigned_integer_format = "%" , [ "+" | "-" ] ,
                     [  "*" , decimal_with_optional_leading_zero ] , "u" ;

signed_hexadcimal_format = "%" , [ "+" | "-" ] ,
                     [  "*" , decimal_with_optional_leading_zero ] , "x" ;

character_array_format = "%" ,
                     [  "*" , decimal_with_optional_leading_zero ] , "c" ;

character_string_format = "%" ,
                     [  "*" , decimal_with_optional_leading_zero ] , "s" ;

defined_character_set_format = "%" ,
                     [  "*" , decimal_with_optional_leading_zero ] ,
                     "[", { ascii - "]" } , "]" ;

real_number_format = "%" , [ "+" | "-" ] ,
                     [  "*" , decimal_with_optional_leading_zero ] ,
                     ( "a" | "e" | "f" | "g" );
```

© ISO 2007 – All rights reserved

```
object_pointer_format = "%" , "p" ;

null_conversion = "%" , "n" ;
```


© ISO 2007 – All rights reserved