

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Multimedia systems and equipment – Multimedia e-publishing and e-books –
Interchange format for e-dictionaries**

**Systèmes et équipements multimédia – Publication et livres électroniques
multimédia – Format d'échange pour les dictionnaires électroniques**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2011 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester.

If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de la CEI ou du Comité national de la CEI du pays du demandeur.

Si vous avez des questions sur le copyright de la CEI ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de la CEI de votre pays de résidence.

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland
Email: inmail@iec.ch
Web: www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

- Catalogue of IEC publications: www.iec.ch/searchpub

The IEC on-line Catalogue enables you to search by a variety of criteria (reference number, text, technical committee,...). It also gives information on projects, withdrawn and replaced publications.

- IEC Just Published: www.iec.ch/online_news/justpub

Stay up to date on all new IEC publications. Just Published details twice a month all new publications released. Available on-line and also by email.

- Electropedia: www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 20 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary online.

- Customer Service Centre: www.iec.ch/webstore/custserv

If you wish to give us your feedback on this publication or need further assistance, please visit the Customer Service Centre FAQ or contact us:

Email: csc@iec.ch

Tel.: +41 22 919 02 11

Fax: +41 22 919 03 00

A propos de la CEI

La Commission Electrotechnique Internationale (CEI) est la première organisation mondiale qui élabore et publie des normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications CEI

Le contenu technique des publications de la CEI est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

- Catalogue des publications de la CEI: www.iec.ch/searchpub/cur_fut-f.htm

Le Catalogue en-ligne de la CEI vous permet d'effectuer des recherches en utilisant différents critères (numéro de référence, texte, comité d'études,...). Il donne aussi des informations sur les projets et les publications retirées ou remplacées.

- Just Published CEI: www.iec.ch/online_news/justpub

Restez informé sur les nouvelles publications de la CEI. Just Published détaille deux fois par mois les nouvelles publications parues. Disponible en-ligne et aussi par email.

- Electropedia: www.electropedia.org

Le premier dictionnaire en ligne au monde de termes électroniques et électriques. Il contient plus de 20 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans les langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International en ligne.

- Service Clients: www.iec.ch/webstore/custserv/custserv_entry-f.htm

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions, visitez le FAQ du Service clients ou contactez-nous:

Email: csc@iec.ch

Tél.: +41 22 919 02 11

Fax: +41 22 919 03 00

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Multimedia systems and equipment – Multimedia e-publishing and e-books –
Interchange format for e-dictionaries**

**Systèmes et équipements multimédia – Publication et livres électroniques
multimédia – Format d'échange pour les dictionnaires électroniques**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

PRICE CODE
CODE PRIX

XF

ICS 33.160.60; 35.240.20; 35.240.30

ISBN 978-2-88912-552-4

CONTENTS

FOREWORD.....	5
INTRODUCTION.....	7
1 Scope.....	8
2 Normative references	8
3 Terms and definitions	8
4 Position and requirements for interchange format for E-dictionaries	8
4.1 Interchange format for e-dictionaries in contents creation/distribution model	8
4.2 Requirements for interchange format for e-dictionaries.....	9
5 File format.....	9
6 Semantics	10
7 File format details.....	10
8 Overview of the format's structure	10
9 Elements and attributes.....	11
9.1 General.....	11
9.2 Page_ID.....	11
9.3 Object_ID.....	11
9.4 Char_ID.....	11
9.5 Reading.....	11
9.6 Filename.....	12
9.7 Standard character.....	12
9.8 Standard character string.....	12
9.9 Extended character.....	13
9.10 Extended character string.....	13
9.11 External character.....	13
9.12 External character string.....	14
9.13 External extended character string.....	15
9.14 Coordinates.....	15
9.15 Polygonal_region.....	15
9.16 Color.....	15
9.17 Date.....	16
9.18 Time.....	16
9.19 Country.....	16
9.20 Personal_name.....	17
9.21 Organization_name.....	17
9.22 Address.....	17
9.23 Permission.....	18
9.24 Keyword.....	19
9.25 Telephone_number.....	19
9.26 Mail_address.....	20
10 Description format details.....	21
10.1 General.....	21
10.2 Book information modules <bvf>.....	21
10.2.1 General.....	21
10.2.2 Bibliographical data <book_info>.....	22
10.3 Content management module <body_module>.....	28

10.3.1	General	28
10.3.2	Flowing content data <flow_type_body>	28
10.4	Event info module <event_info>	41
10.4.1	Events	41
10.4.2	Event data <event>	41
10.4.3	Trigger (pointer) <trigger_pointer>	42
10.4.4	Action information	43
10.5	Parts data module <parts_module>	45
10.5.1	Storage and management	45
10.5.2	Dynamic text object <dynamic_text_object_entry>	46
10.5.3	Sound object <sound_object_entry>	47
10.5.4	Search page object <search_page_object_entry>	47
10.5.5	Movie object <movie_object_entry>	48
10.5.6	Dictionary data object <dict_data_object_entry>	48
10.6	Object instances	49
10.6.1	General	49
10.6.2	Text object instance <text_data>	49
10.6.3	Dictionary data object instance	71
10.6.4	Image object instance	77
10.6.5	Sound object instance	77
10.6.6	Animation object instance	77
10.6.7	Search page object instance <search_page>	78
10.6.8	Movie object instance	83
Annex A	(normative) Color names	84
Bibliography		126
Figure 1	– Contents creation/distribution model	8
Figure 2	– Contents creation/distribution model (modified)	9
Figure 3	– Relationship between concepts	9
Figure 4	– XML tree structure	10
Figure 5	– Example of valign="middle"	50
Figure 6	– Example of dropped capital	54
Figure 7	– Left and right margin of a paragraph	55
Figure 8	– Horizontal writing in vertical text	60
Figure 9	– Ruby	61
Figure 10	– Example of search page object instance rendering	80
Table 1	– Base characters for reading	12
Table 2	– Standard character set	12
Table 3	– Usable characters for a telephone number	20
Table 4	– Characters usable for email addresses	20
Table 5	– Characters usable for the lookup key	40
Table A.1	– Color names	84
Table A.2	– Examples of additional standard character sets	85
Table A.3	– Example of additional characters usable for readings	85
Table A.4	– Example of additional sorting rules	87

Table A.5 – Example of additional language specific attributes for <enable_key_type> 88

Withdawn
IECNORM.COM: Click to view the full PDF of IEC 62605:2011

INTERNATIONAL ELECTROTECHNICAL COMMISSION

MULTIMEDIA SYSTEMS AND EQUIPMENT – MULTIMEDIA E-PUBLISHING AND E-BOOKS – INTERCHANGE FORMAT FOR E-DICTIONARIES

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

International Standard IEC 62605 has been prepared by technical area 10: Multimedia e-publishing and e-book, of IEC technical committee 100: Audio, video and multimedia systems and equipment.

The text of this standard is based on the following documents:

FDIS	Report on voting
100/1829/FDIS	100/1863/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

The committee has decided that the contents of this publication will remain unchanged until stability date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

Withdrawn
IECNORM.COM: Click to view the full PDF of IEC 62605:2011

INTRODUCTION

Markets for multimedia e-book and e-publishing require standardization of formats for e-book data interchange among associated people; authors, data preparers, publishers and readers. The formats are classified into submission format, interchange format and reader's format. The submission format supports an interaction between authors and data preparers. The reader's format depends on e-publishing equipment. The interchange format provides an interchange format for data preparers and publishers and therefore should be e-publishing equipment independent.

The International Electrotechnical Commission (IEC) draws attention to the fact that it is claimed that compliance with this document may involve the use of patents.

Sharp Corporation has informed IEC that it has patent applications or granted patents.

The IEC takes no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured the IEC that he/she is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with the IEC. Information may be obtained from:

Sharp Corporation
Platform Technology Development Center
2613-1, Ichinomoto-cho, Tenri, Nara
632-8587 JAPAN

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. The IEC shall not be held responsible for identifying any or all such patent rights.

ISO (www.iso.org/patents) and IEC (<http://patents.iec.ch/>) maintain on-line data bases of patents relevant to their standards. Users are encouraged to consult the data bases for the most up to date information concerning patents.

MULTIMEDIA SYSTEMS AND EQUIPMENT – MULTIMEDIA E-PUBLISHING AND E-BOOKS – INTERCHANGE FORMAT FOR E-DICTIONARIES

1 Scope

This International Standard specifies the interchange format for e-dictionaries among publishers, content creators and manufacturers.

This International Standard does not address the following aspects:

- data formats for reading devices;
- elements necessary for final print reproduction only;
- rendering issues related to physical devices;
- security issues such as DRM for document.

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.

IEC/TS 62229:2006, *Multimedia systems and equipment – Multimedia e-publishing and e-book – Conceptual model for multimedia e-publishing*

3 Terms and definitions

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

3.1

manufacturer

organization or person that manufactures hardware and/or software of the e-book

4 Position and requirements for interchange format for E-dictionaries

4.1 Interchange format for e-dictionaries in contents creation/distribution model

The conceptual model for multimedia e-publishing (IEC/TS 62229) defines a contents creation/distribution model shown in Figure 1.

Author <--(1)--> Data preparer <--(2)--> Publisher --(3)--> Reader

IEC 1383/11

Key

- (1) content data in submission format
- (2) content data in interchange format
- (3) content data in reader's format

Figure 1 – Contents creation/distribution model

It should be noted that the role of manufacturers of e-dictionary hardware and software overlaps that of the publisher in Figure 1. Therefore, a slightly modified model will be assumed for this International Standard, as shown in Figure 2.

Author <--(1)--> Data preparer <--(2)--> Publisher (manufacturer) --(3)--> Reader

IEC 1384/11

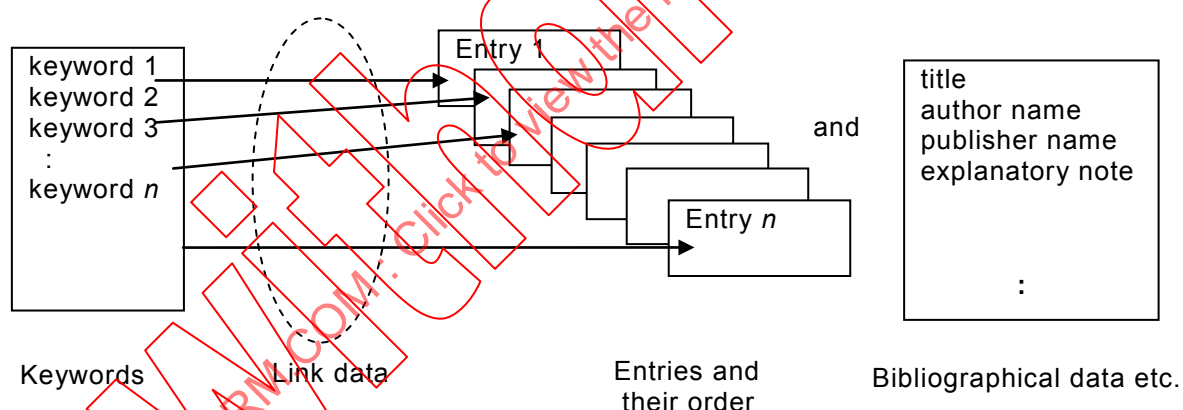
Figure 2 – Contents creation/distribution model (modified)

This International Standard specifies the interchange format between data preparers and publishers, i.e. a format for (2) in Figure 2 though it may be used as a reader's format.

4.2 Requirements for interchange format for e-dictionaries

An interchange format for e-dictionaries needs to address the following.

- Description of keywords, links from the keywords to entries (link data) and the order of the entries.
- Description of articles for each entry (entry data). This includes text, image, and other multimedia functionalities generally required for e-books.
- Description of bibliographical data and other data. This should include the name of the author and the publisher, the title of the content and the explanatory note. The relationship between these concepts is visually represented in Figure 3.
- Description of contents written in various languages.



IEC 1385/11

Figure 3 – Relationship between concepts

5 File format

This International Standard is based on XMDF (as described in IEC 62448:2009, Annex B) and LeXML. The standard format is represented in XML and hereafter called XMDF-LeXML format.

NOTE LeXML is proposed by Digital ASSIST Ltd. Its original specifications are found at <http://www.d-assist.com/index.html>.

6 Semantics

Elements of the XMDF-LeXML format can be rendered in accordance with appropriate style specifications, which are outside the scope of this International Standard.

7 File format details

This Clause specifies the XMDF-LeXML format mentioned in Clause 5.

The XMDF-LeXML format is an interchange format for e-dictionaries multimedia e-book data interchange, targeted at data preparers and publishers rather than the reader, with an emphasis on mobile devices as a target platform. Much like HTML, this format does not split the document in fixed pages, but determines the layout according to the viewer device's display size, the font in use, and so on. In this standard, such contents will be referred to as flowing content, as opposed to paged content.

8 Overview of the format's structure

Flowing contents are usually composed of several concatenated flows. This standard makes no particular requirement concerning the way the flowing content should be split into individual flows. This decision is left to the data preparer, to accommodate the various type of contents. For instance, a newspaper may have one flow per article, a novel one per chapter, and so on. It is also possible not to split the content, and to have only one flow. However, it should be noted that particularly large flows, or an extremely large number of flows, may impact on runtime performance, depending on the specific version of the viewer in use, the available memory, and so on.

The XML tree structure of the format is shown in Figure 4.

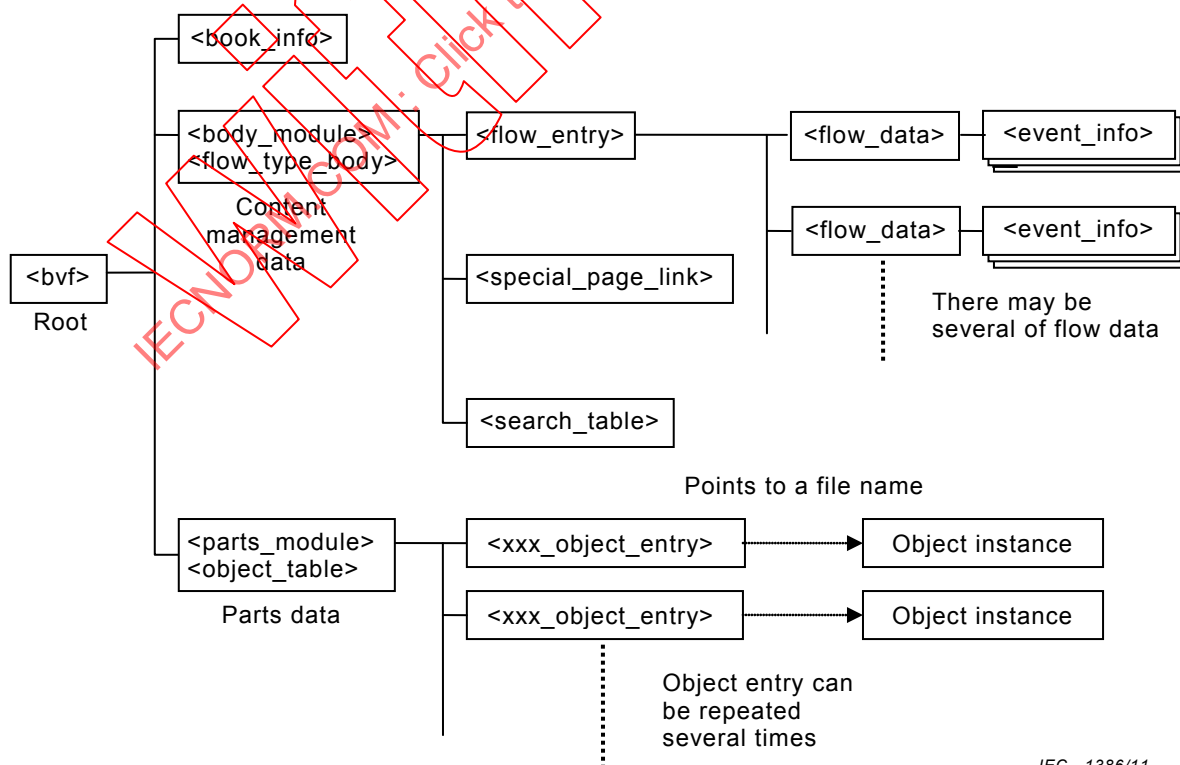


Figure 4 – XML tree structure

The actual contents of each flow, in other words, what will be displayed by the viewer, is recorded in the *object instance*. The *object instance* is registered in *object_entry*, and associated with an ID number and other auxiliary data, turning it into playable / displayable data. *Flow_data* determines its content by pointing at such registered objects. In addition, information on functionalities such as page link is recorded in *event_info*.

The main part of this standard is generic, and may be used for any country and language. However, some parts may have language specific behavior. To keep the main part of the specification reasonably light and focused, those localization-related issues are detailed in Clause A.2. The main text will refer to them, when appropriate.

9 Elements and attributes

9.1 General

The different types of values that may be used in the various tags or attributes are explained below. The elements and attributes detailed below will be valid throughout this standard, and will be referenced by other constructs. In the following explanations, alphanumeric characters refer to numerals from 0 to 9 and alphabetic letters from a to z and A to Z.

9.2 Page_ID

Page_ID specifies a unique identification number for the flow data of the flowing contents. It is a string starting by the "PG" characters, followed by alphanumeric characters.

Example:

```
<flow_data flow_id="PG0002" ... />
```

9.3 Object_ID

Object_ID specifies a unique identification number for objects used in the flowing contents. It is a string starting by the "OB" characters, followed by alphanumeric characters.

Example:

```
<dynamic_text_object_entry id="OB0ue4" ... />
```

9.4 Char_ID

Char_ID specifies an identification number for positions (character strings, etc.) within text and dictionary data objects. It is an alphanumeric string which is to be given uniquely in the *text* (see 10.6.2) and *dictionary data object instance* (see 10.6.3). Char IDs with the same value in different object instances are regarded as separate and don't affect each other.

Example:

```
<trigger_pointer id="OB29s0/CR0de4"/>  
Click<char_id char_id="CR0001">here</char_id>for details.
```

9.5 Reading

For sorting purposes, reading may be useful to specify the reading of each word. Restricting the characters allowed for this purpose to a limited set makes it easier to define the sorting method. Such characters should be determined on a per language basis. All languages can

use the characters listed in Table 1 as a common base, while the localization (see Clause A.2) will describe the language specific extensions to it.

Table 1 – Base characters for reading

Name	Corresponding characters ^{a)}
Basic alphabet	A to Z (0x0041 to 0x005A) A to z (0x0061 to 0x007A)
Numerals	0 to 9 (0x0030 to 0x0039)
Others	space (0x0020), ((0x0028),) (0x0029)
^{a)} All values are in Unicode.	

Example:

```
<title reading="PI">π</title>
```

9.6 Filename

Filenames should be written using the following convention. The path is relative to the file in which this reference is made. Network paths should not be used. For portability concerns, it is recommended that only ACSII characters be used. Both the slash and backslash characters are acceptable as directory separators. It is also recommended not to use excessively long filenames, as those might not be supported by the host operating system.

Example:

```
<dynamic text object entry src="sect1.xml" type="text/x-bvf-text" id="OB0ue4"/>
```

9.7 Standard character

The standard character set of the document, as set by the default_ccs attribute of the <bvf> tag (see 10.2), is to be chosen from a well defined list, so as to ease the development of viewing software. However, this list may change for different localized versions of the XMDF-LeXML format. Any e-book data has to define its standard character set as one of or a combination of character set(s) listed in Table 2 and those defined for a specific localization (see A.2.2).

Table 2 – Standard character set

Character set name	Description
"ISO 646-IRV"	Characters in the range of US-ASCII

9.8 Standard character string

A string composed of *Standard characters* is called a *Standard character string*. Unless specified otherwise, the spacing characters (space (0x0020), linefeed (0x000D, 0x000A, 0x0D0A), tabulation (0x0009)) are to be handled as follows:

Space (0x0020) to be displayed as is.

Linefeed (0x000D, 0x000a, 0x0D0A) not to be displayed, but simply ignored.

Tabulation (0x0009) to be displayed as if it were a single space.

Furthermore, because of restriction in the XML format, linefeeds (0x000D, 0x000A, 0x0D0A) and tabulations (0x0009) in attribute values should be replaced by spaces when converting to the distribution format.

9.9 Extended character

Characters which have Unicode code points while not being among those listed below are referred to as *Extended characters*.

Standard characters

Surrogate pair range (0xD800 to 0xDFFF)

BOM (Byte Order Mark) (0xFFFE, 0xFEFF)

NON CHARACTER (0xFFFF)

Control characters (characters between 0x0000 and 0x001F except tabulation (0x0009) and linefeed (0x000A, 0x000D), as well as DEL (0x007F)).

If an e-book indeed uses any *Extended character* in its data, the name of a charset covering those *Extended characters* should be appended to the *default_ccs* attribute of the <bvf> tag. Note that all *Extended characters* used in the document do not need to be covered by the same character set, as it is possible to specify several ones.

9.10 Extended character string

A string composed of *Standard characters* and *Extended characters* is called an *Extended character string*. Unless specified otherwise, the spacing characters (space (0x0020), linefeed (0x000D, 0x000A, 0x0D0A), tabulation (0x0009)) are to be handled the same way as in *Standard character strings*.

9.11 External character

To display a character which is neither a *Standard character* nor an *Extended character*, it is possible to use the <external_char> tag described below.

<external_char> inserts an *External character*. The viewer may display it according to the following methods.

- Display the character set by the alt_set and alt_code attributes.
- Display the image set by the alt_img or alt_vimg attributes.
- Display the alternative letter set by the alt attribute.

It has the following attributes.

[Attributes]

alt_set: Together with the alt_code attribute, it allows designating the *External character* to be used. This alt_set attribute indicates the font name, while the alt_code attribute indicates the character code point within the font. The alt_set attribute is written in the following way:

alt_set = "font1,font2, ..."

The alt_set attribute may hold several font names, separated by "," (0x002C). In that case, the viewer should use the first font of the list that is available (either from the platform, or included in the contents data itself) to display the character.

alt_code: Selects a character code point in the font specified by the alt_set attribute. It may be written both as a decimal number, or a hexadecimal number, prefixed by "0x". In case several fonts have been defined in the alt_set attribute, the character code shall represent the same character in all of them. This attribute can be omitted.

alt_img: Defines an alternative character image. Written as a *Filename*. Before opening the file indicated by this attribute, the img_type attribute should be checked for authorized file types. Note that it may be used only when

<external_char> is used in a *text object instance*. When both alt_img and alt_vimg are used, the file types shall match. This can be omitted. When this attribute is set, display should be done according to the following methods:

a) Monochrome images

Black pixels represent the letter, and white pixels, the background. The font color and background color are to be displayed according to the color attribute of the tag.

b) Images with levels of gray

Black pixels represent the letter, and white pixels, the background. The font color and background color are to be displayed according to the color attribute of the tag. The color of "gray" pixels shall be computed as an intermediate value between the font color and the background color.

c) Color images

Displayed as is.

alt_vimg: Defines an alternative character image to be used when the text is displayed vertically (as can be the case in some languages, such as Japanese). Written as a *Filename*. When omitted, the image defined in the alt_img attribute should be used both for horizontal and vertical layout. Before opening the file indicated by this attribute, the img_type attribute should be checked for authorized file types. Note that it may be used only when <external_char> is used in a *text object instance*. When both alt_img and alt_vimg are used, the file types shall match. This can be omitted. When this attribute is set, the display should be done according to the same methods as with alt_img.

img_type: Defines the mime type of the images files set in the alt_img and alt_vimg attributes. Currently, only PNG and JPEG are supported, and should be written as:

"image/png"

"image/jpeg"

When either alt_img, alt_vimg or both are set, this attribute is required. As these two, it can only be used in an <external_char> tag inside a *text object instance*.

alt: Alternative character string. Written as a *Standard character string*. May be omitted.

Example:

```
<external_char alt_set="oooextchars" alt_code="47268" alt_img="ou.img"
alt_vimg="ou_v.img" img_type="image/jpeg" alt="鷗"/>
<external_char alt="間"/>
<external_char alt_set="sharp_extchars" alt_code="0x2345" alt="間"/>
```

9.12 External character string

An *External character string* is a string composed of *Standard characters*, *External characters*, or *both*. Unless specified otherwise, spacing characters (space (0x0020), linefeed (0x000D, 0x000A), tabulation (0x0009)) should be handled the same way as they are handled in *Standard character strings*.

Example:

```
森<external_char alt_img="ou.png" alt_type="image/png" alt="鷗"/>外  
内田百 <external_char alt_set="sharp_extchars" alt_code="0x2345" alt="間"/>
```

9.13 External extended character string

An *External extended character string* is a string of *Standard characters*, *Extended characters*, *External characters*, or a combination of any of the above. Unless specified otherwise, spacing characters (space (0x0020), linefeed (0x000D, 0x000A), tabulation (0x0009)) should be handled the same way as they are handled in *Standard character strings*.

9.14 Coordinates

Data type to be used to store coordinates, dimension and other similar information composed of an x and a y value. It is written as "(x, y)". The name of the attribute which uses this type of data depends on the tag.

The coordinate system explained below will be used in this standard. The origin is at the top left corner, the x axis oriented rightwards, and the y axis downwards. As the system of coordinates used by the viewer to map things on the screen is implementation-dependent, it will not be discussed here.

Local coordinate system:

The coordinate system local to an object takes its origin in the upper left corner of the circumscribed rectangle, and has the same orientation as the general coordinate system. Positions within an object should be expressed in the local coordinate system.

Example:

```
<vertex position="(100,200)"/>
```

9.15 Polygonal_region

Data format to store the apexes of a polygon, or any other ordered sequence of vertexes. Each vertex is stored in a <vertex> tag. When defining the shape of a polygon, the edges shall not cross. If they do, the viewer's behavior is unspecified. The <vertex> tag has the following attribute.

[Attribute]

position: the position of the apex, expressed as *Coordinates*. This attribute Shall not be omitted.

Example:

```
<vertex position="(100,0)"/> <!-- in the case of a triangle -->  
<vertex position="(0,100)"/>  
<vertex position="(200,100)"/>
```

9.16 Color

Data type to define colors. The following attributes are defined.

[Attributes]

- color_space:** Specifies the color space to be used. Currently, only RGB is accepted. If this attribute is omitted, the viewer should act as if RGB was set.
- color:** Specifies the color name. Color names or numerical values may be used. Acceptable color names are listed in Table A.1. The default value depends on the actual tag and context. Numerical values are to be written in the following syntax.
- In RGB: written as #RRGGBB. With RR, GG, BB being hexadecimal numbers, ranging from 00 to FF. Grayscale pixel values are represented by setting RR, GG and BB to the same value.
- opacity:** level of opacity. Ranging from 0 (transparent) to 100 (opaque). Presently, the only admitted value is 100, and in case the attribute is omitted, it defaults to 100.

Example:

```
<font color="#FF0000"/>
    <!-- the color_space is unspecified, and thus defaults to RGB-->
<font color="#FF0000" opacity="100"/>
<font color="black"/>
```

9.17 Date

Data format to store dates. It uses the same representation as ISO 8601. For instance, 1994-11-05T08:15:30-05:00 corresponds to November 5, 1994, 8:15:30 am, US Eastern Standard Time. Abbreviated forms are also accepted. Please refer to <http://www.w3.org/TR/NOTE-datetime> for details.

Example:

```
<publication date type="publish">1994</publication date>
```

9.18 Time

Data format to specify durations. Written as "XXdXXhXXmXXsXXXms", where X stands for a digit between 0 and 9. For instance, "10d5h30m10s015ms" would mean 10 days, 5 hours, 30 minutes, 10 seconds and 15 milliseconds. Abbreviated forms such as "5m30ms" or "1s" are possible. There is no upper bound to the number of days.

Example:

```
<flip animation renewal time="1s" >
```

9.19 Country

Data format to specify a country name. Written according to the ISO 3166-1 alpha 3 standard, in lower case.

Example:

```
<publication place>jpn</publication place>
```

9.20 Personal_name

Data format to store people's names, such as the document author. It is stored under the <personal_name> tag. Several child elements are used to define the various parts of the name: first name, middle name and last name. This information shall be entered via the tags defined below. At least one of <first_name>, <middle_name>, and <last_name> shall be specified.

[Child elements]

<first_name> Sets the first name. Written as an *External character string*. May be omitted. Accepts the following attribute.

[Attribute]

reading: Pronunciation of the first name, written as a *Reading*. May be omitted.

<middle_name> Sets the middle name. Written as an *External character string*. May be omitted. Accepts the following attribute.

[Attribute]

reading: Pronunciation of the middle name, written as a *Reading*. May be omitted.

<last_name> Sets the last name. Written as an *External character string*. May be omitted. Accepts the following attribute.

[Attribute]

reading: Pronunciation of the last name, written as a *Reading*. May be omitted.

Example:

```
<personal_name>
  <!-- to record John Smith -->
  <last_name>Smith</last_name>
  <first_name>John</first_name>
</personal_name>
```

9.21 Organization_name

Data format to define company's name, such as publishers. It is stored in the <organization_name> tag. Written as an *External character string*. The following attribute may be specified.

[Attribute]

reading: Pronunciation of the company's name, written as a *Reading*. May be omitted.

Example:

```
<organization_name >ABCD Corporation</organization_name>
```

9.22 Address

Data format used to define an address, telephone number, email, and other information. Everything is stored in child elements of a main <address_info> tag.

[Child elements]

- `<postal_code>` Stores the postal code (zip code) as a *Standard character string*. May be omitted.
- `<address>` Stores the address as an *External Character String*. May be omitted.
- `<telephone>` Stores the phone number as a *Standard character string*. May be omitted.
- `<fax>` Stores the fax number as a *Standard character string*. May be omitted.
- `<mail_address>` Stores the mail address as a *Standard character string*. May be omitted.
- `<website>` Stores the home page's URI as a *Standard character string*. May be omitted.
- `<address_other_info>` Allows storing additional information not covered by the preceding elements as an *External character string*. May be omitted. If this information is to be displayed, spaces (0x0020) and linefeeds (0x000D, 0x000A, 0x0D0A) should be displayed as is, while tabulations (0x0009) should be displayed as spaces.

Example:

```
<address_info>
  <postal_code>75008</postal_code>
  <address>xxx avenue des champs elysees. Paris, France </address>
  <telephone>01 53 xx xx xx</telephone>
  <mail_address>xxx@XXXXXX.fr</mail_address>
  <website>http://www.XXXXXX.fr</website>
</address_info>
```

9.23 Permission

Sets the permissions, such as the right to print, or copy. The various permissions are stored in child elements of the `<permission_info>` tag. When the `<permission_info>` tag is omitted, all permissions are set to the same value as when each permission tag is omitted. The following explanations refer to an "authenticated user". Under usual circumstances, all viewers of the document are considered authenticated. However, the distribution format may include DRM technologies and authentication mechanisms. The following child elements are available.

[Child elements]

- `<print_permission>` Defines whether printing is permitted or not. When omitted, it is to be considered as if the permission attribute was set to "no".

[Attribute]

- permission: Defines whether printing is permitted or not. The following values are possible.
 - "authorized": authenticated users may print.
 - "no": no one can print (default value).

- `<copy_permission>` Define whether copying is permitted or not. When omitted, it is to be considered as if the permission attribute was set to no.

[Attribute]

- permission: Defines whether copying is permitted or not. The following values are possible.
 - "authorized": authenticated users may copy.

"in_device_only": authenticated users may copy, but only within the viewer device. If the device does not provide mechanisms to prevent external copy, then copy is forbidden.

"no": No one can copy (default value).

Example:

```
<permission_info>
  <print_permission permission="authorized"/>
</permission_info>
```

9.24 Keyword

With the <keyword_list> tag, it is possible to attach a list of keywords to the bibliographical data, to a flow data, or to an object (in the present specification, it is limited to bibliographical data). A <keyword_list> should contain one or more <keyword> tags as child element, each of these recording one keyword. The <keyword> tag has the following attributes and child element.

[Attributes]

category: Defines the category the keyword belongs to, written as a *Standard character string*. May be omitted.

reading: Records the pronunciation of the keyword, written as a *Reading*. May be omitted.

[Child elements]

External character string: Records the actual keyword.

Example:

```
<keyword_list>
  <keyword category="History" >Renaissance</keyword>
  <keyword>xml</keyword>
</keyword_list>
```

9.25 Telephone_number

Data format to record telephone numbers. It allows dialing such a number when the viewer device is a telephone. It is written as a combination of the characters listed in Table 3. The number of characters shall be between 1 and 64, inclusive.

If a function defined by the phone number cannot be executed, the viewer should not make the call. Namely,

when the phone number string is too long for the device to handle,
 when the phone number string includes characters not listed in Table 3,
 when the phone number string contains a function which cannot be executed.

Table 3 – Usable characters for a telephone number

Character	ASCII code	Meaning
0 to 9	0x30 to 0x39	number
#	0x23	# button
*	0x2A	* button
-	0x2D	Ignored.
,	0x2C	Pause (1 second). If a 1 second pause cannot be made, wait for key press instead.
/	0x2F	Pause (wait for key press).
P or p	0x50 or 0x70	Pause (wait for signal). If it is not possible to wait for the signal, wait for a key press instead.
+	0x2B	Sign to make an international call. Only at the beginning of a phone number. (If entered in another position, do not make the call.)

9.26 Mail_address

Data format to store an email address. Written as local-part@domain. local-part and domain may use any of the characters recorded in Table 4. The maximum length for local-part@domain is 256 bytes. Within the following characters, "&" (0x26) shall be written as an xml entity: "&". Such xml entities are counted as one byte.

Table 4 – Characters usable for email addresses

Category	Characters	ASCII code
numerals	0 to 9	0x30 to 0x39
alphabet	A to Z	0x41 to 0x5A
	a to z	0x61 to 0x7A
!	!	0x21
\$	\$	0x24
%	%	0x25
&	&	0x26
*	*	0x2A
+	+	0x2B
-	-	0x2D
.	.	0x2E
/	/	0x2F
=	=	0x3D
?	?	0x3F
^	^	0x5E
_	_	0x5F
~	~	0x7E

10 Description format details

10.1 General

Each book takes the form of an xml document as can be seen below.

```
<?xml version="1.0" encoding="UTF-8" ?>
<bvf id="1234" id-type="...">
    <book_info>
        ...      <!-- Bibliography comes here -->
    </book_info>

    <body_module>
        ...      <!-- Flow data comes here -->
    </body_module>

    <parts_module>
        ...      <!-- Objects are registered here -->
    </parts_module>
</bvf>
```

The character encoding used in the document is specified in the usual XML way, with the encoding attribute of the <xml> tag. This standard recommends using UTF-8 or UTF-16 to avoid conversion problems, see A.2.8 for further details.

10.2 Book information modules <bvf>

10.2.1 General

The book information module is recorded in the <bvf> tag. It serves as a root tag for all data in the book, all information related to the document are stored inside it (it may also happen that only the filename of external files is stored here).

The <bvf> tag has the following attributes and child elements.

[Attributes]

- id_type: Defines what type of number is stored in the id attribute. Written as a *Standard character string*. May be omitted.
- id: Records the Identification number of this book, in the system specified by the id_type attribute. Written as a *Standard character string*. May be omitted. Not to be confused with Reference ID (see 10.6.3.4).
- default_ccs: Sets the name of the character set of the *Standard characters* and *Extended characters* used in this standard (see Clause 9). When more than one character sets are specified, they are separated by a "," (0x002C). Shall not be omitted.
- display_size: Specifies the display (screen) size that was assumed while creating the contents, written in the standard *coordinates* format. May be omitted.

[Child elements]

- <book_info> Records bibliographical data. Shall not be omitted. See 10.2.2 for details.
- <body_module> Content management module. Shall not be omitted. See 10.3 for details.
- <parts_module> Parts data modules. Shall not be omitted. See 10.5 for details.

Example:

```
<bvf id_type="ISBN" id="xxx-x-xxxx-xxxx-x" default_ccs="ISO 646-IRV">

    <book_info> ... </book_info>          <!-- Bibliography -->
    <body_module> ... </body_module>      <!-- Content module -->
    <parts_module> ... </parts_module>    <!-- Parts data modules -->

</bvf>
```

10.2.2 Bibliographical data <book_info>

This is where bibliographical data, such as the author or the title, is stored. This <book_info> tag has the following child elements.

[Child elements]

<title_info> Stores the information related to the title. Shall not be omitted. It has the following child elements.

[Child elements]

<series_title> The title of the series is recorded as an *External character string* in this element. If there is no series' title, it may be omitted. The following attribute may be used.

[Attribute]

reading: gives the pronunciation of the series' title as a *Reading*. May be omitted.

<title> The title is recorded as an *External character string* in this element. It shall not be omitted. The following attribute may be used.

[Attribute]

reading: gives the pronunciation of the title as a *Reading*. May be omitted.

<subtitle> The subtitle is recorded as an *External character string* in this element. If there is no subtitle, it may be omitted. The following attribute may be used.

[Attribute]

reading: gives the pronunciation of the subtitle as a *Reading*. May be omitted.

<edition_info> Information concerning the revision history of the book is recorded as an *External character string* in this element. May be omitted. If the information is to be displayed, the space character (0x0020), and the linefeed and carriage return characters (0x000D, 0x000A, 0x0D0A) are to be displayed as is, but the tabulation character (0x0009) is to be displayed as if it was a space. The following attribute may be used.

[Attribute]

this_version: Specifies which of the cited versions is the present one. For instance, "Third revision". May be omitted.

<title_other_info> Other information related to the title may be stored as an *External character string* in this element. May be omitted. If the information is to be displayed, the space character (0x0020), and the linefeed and carriage return characters (0x000D, 0x000A, 0x0D0A) are to be displayed as is, but the tabulation character (0x0009) is to be displayed as if it were a space.

<author_info> Stores the information related to the author. May be omitted. Each author is registered in a separate <author> child element.

The <author> tag has the following attribute and child elements.

[Attribute]

role: defines the role of the person mentioned. The possible values are listed below. If omitted, it defaults to "author".
 "author", "editor", "translator", "supervisor", "designer",
 "photographer", "illustrator", "binder", "planner", "other"

[Child elements]

<personal_name> / <organization_name>

Records the author's name in one of these two tags, according to whether the author is an individual or an organization, respectively written as a *Personal_name* or *Organization_name*.

<address_info> Address of the author, written in the *Address* data format. May be omitted.

<author_other_info> Other information related to the author may be stored as an *External character string* in this element. May be omitted. If the information is to be displayed, the space character (0x0020), and the linefeed and carriage return characters (0x000D, 0x000A, 0x0D0D) are to be displayed as is, but the tabulation character (0x0009) is to be displayed as if it were a space.

<publisher_info> Stores the information related to the publisher. May be omitted. The <publisher_info> tag has the following child elements. If <publisher_info> is specified, at least one of <publisher> or <publisher_office> shall be specified as well.

[Child elements]

<publisher> Stores the information about the publisher if it is an individual. May be omitted. It has the following child elements.

[Child elements]

<publisher_name> The name of the publisher is recorded as an *External character string* in this element. Shall not be omitted. The following attribute may be used.

[Attribute]

reading: gives the pronunciation of the publisher's name as a *Reading*. May be omitted.

<address_info> Stores the address of the publisher, in the standard *Address* format. May be omitted.

<publisher_office> Stores the information about the publisher if it is a company. May be omitted. It has the following child elements and attributes.

[Attribute]

publisher_code: Records the publisher's ID. May be omitted.

[Child elements]

<organization_name> The organization's name is recorded in the element. May be omitted.

<address_info> The publisher's address is recorded in this element, in the standard *Address* format. May be omitted.

<publisher_other_info> Other information related to the publisher may be stored as an *External character string* in this element. May be omitted. If the information is to be displayed, the space character (0x0020), and the linefeed and carriage return characters (0x000D, 0x000A, 0x0D0A) are to be displayed as is, but the tabulation character (0x0009) is to be displayed as if it were a space.

<seller_info> Stores the information related to the seller. May be omitted. The <seller_info> tag has the following child elements. If <seller_info> is specified, at least one of <seller> or <seller_office> shall be specified as well.

[Child elements]

<seller> Stores the information about the seller if it is an individual. May be omitted. It has the following child elements.

[Child elements]

<seller_name> The name of the seller is recorded as an *External character string* in this element. Shall not be omitted. The following attribute may be used.

[Attribute]

reading: gives the pronunciation of the seller's name as a *Reading*. May be omitted.

<address_info> Stores the address of the seller, in the standard *Address* format. May be omitted.

<seller_office> Stores the information about the seller if it is a company. May be omitted. It has the following child elements and attributes.

[Attribute]

seller_code: Records the seller's ID. May be omitted.

[Child elements]

<organization_name> Records the organization's name. May be omitted.

<address_info> Records the seller's address, written in the standard *Address* format. May be omitted.

<seller_other_info> Other information related to the seller may be stored as an *External character string* in this element. May be omitted. If the information is to be displayed, the space character (0x0020), and the linefeed and carriage return characters (0x000D, 0x000A, 0x0D0A) are to be displayed as is, but the tabulation character (0x0009) is to be displayed as if it were a space.

<book_id_info> Records the books identification number, such as its ISBN number. May be omitted. It has the following child element.

[Child elements]

<book_id> Each type of identification number is stored in a <book_id> tag, written as a *Standard character string*. When <book_id_info> is not omitted, there shall be at least one <book_id>. The following attribute shall be set.

[Attribute]

type: Specifies the type of the identification number, such as "ISBN" for instance. Written as a *Standard character string*. Shall not be omitted.

<classification_info> Stores information on the classification of the book. May be omitted. It has the following child element.

[Child elements]

<classification> Each different type of classification is stored in a separate <classification> tag. When <classification_info> is not omitted, there shall be at least one <classification>. It is stored as an *External character string*. The following attribute shall be set.

[Attribute]

type: type of the classification used. Shall not be omitted.

<rating> Allows to rate the contents as violent, or adult. May be omitted. The following attributes may be used.

[Attributes]

adult: Rates the contents as adult oriented materials. Possible values are "yes" or "no". Defaults to no in case of omission.

violence: Rates the contents as violent. Possible values are "yes" or "no". Defaults to no in case of omission.

<publication_place> The country of publication is recorded as a standard *Country* in this element. May be omitted.

<publication_date_info> Stores information regarding the publication date of the book. May be omitted. Each relevant date is to be stored in separate instances of the following child element.

[Child elements]

<publication_date> Stores a date relevant to the publication, such as the publication date itself as well as other dates such as the printing date, or the beginning of sales date, etc. Each date is stored in the standard *Date* format. If <publication_date_info> is not omitted, there shall be at least one <publication_date>. The following attribute may be used.

[Attribute]

type: Specifies what type of date it is. "publish" if it is the publication date and "sale" if it is the beginning of sales date. If omitted, it will default to "publish".

<net_price_info> Defines the price of the book. May be omitted, if the price is open, or not set. More than one prices, classified by currency and country, may be stored in separate instances of the following child element.

[Child elements]

<net_price> Stores a price specific to one country and currency, written as a *Standard character string*. If <net_price_info> is not omitted, there shall be at least one <net_price>. The following attributes may be used.

[Attributes]

country: Defines the country in which this price should apply, in the standard *Country* data format. If omitted, it applies to all countries.

unit: Defines the currency, as a *Standard character string*. Shall not be omitted.

other_info: Other information, written as a *Standard character string*. May be omitted.

<book_abstract> An abstract of the book, written as an *External character string*, is recorded in this element. May be omitted. If the information is to be displayed, the space character (0x0020), and the linefeed and carriage return characters (0x000D, 0x000A, 0x0D0A) are to be displayed as is, but the tabulation character (0x0009) is to be displayed as if it were a space.

<front_cover_image> Defines the image to use as front cover image, by recording its location as a standard *Filename*. Currently, jpeg and png images are supported. May be omitted. The file type as defined by it shall be checked against the type attribute before opening the file. The following attribute shall be used.

[Attribute]

type: Defines the type of the image, by giving its MIME type, for example "image/jpeg". Shall not be omitted.

<spine_cover_image> Define the image to use as a spine image. Follows the same rules as <front_cover_image>. May be omitted.

<keyword_list> Records a list of keywords related to the book's data, written in the *Keyword* syntax. May be omitted

<other_book_info> Other information related to the book may be stored as an *External character string* in this element. May be omitted. If the information is to be displayed, the space character (0x0020), and the linefeed and carriage

return characters (0x000D, 0x000A, 0x0D0A) are to be displayed as is, but the tabulation character (0x0009) is to be displayed as if it were a space.

Example:

```
<book_info>
  <title_info>
    <series_title>Dummy books</series_title>
    <title>the dummy book of nonsense</title>
    <edition_info>2000/01/01 first edition,
                    2005/01/01 second edition</edition_info>
  </title_info>
  <author_info>
    <author role="author">
      <personal_name>
        <first_name>John</first_name>
        <last_name>Smith</last_name>
      </personal_name>
      <address_info>
        <mail_address>john.smith@abcd.com</mail_address>
        <website>http://www.abcd.com/~jsmith/</website>
      </address_info>
    </author>
  </author_info>
  <publisher_info>
    <publisher_office>
      <organization_name>abcd corporation</organization_name>
      <address_info>
        <postal_code>100-1000</postal_code>
        <address>1 main street, Foobar city, Japan</address>
      </address_info>
    </publisher_office>
  </publisher_info>
  <book_id_info>
    <book_id type="ISBN">xxx-x-xxxx-xxxx-x</book_id>
    <book_id type="Japanese_ID_number">454745-7</book_id>
  </book_id_info>
  <classification_info>
    <classification type="Japanese_C_CODE">2143</classification>
  </classification_info>
  <rating adult="no" violence="no"/>
  <publication_place>jpn</publication_place>
  <publication_date_info>
    <publication_date type="publish">2005</publication_date>
  </publication_date_info>
  <net_price_info>
    <net_price country="jpn" unit="yen">1200</net_price>
  </net_price_info>
  <book_abstract>This book doesn't talk about anything special.</book_abstract>
  <front_cover_image type="image/png">xxx.png</front_cover_image>
  <spine_cover_image type="image/png">yyy.png</spine_cover_image>
  <keyword_list>
    <keyword>dummy</keyword>
    <keyword>nonsense</keyword>
  </keyword_list>
</book_info>
```

10.3 Content management module <body_module>

10.3.1 General

The content management module (<body_module>) is in charge of coordinating the contents data into making the actual document. It has the following child element.

[Attribute]

None

[Child elements]

<flow_type_body> Handles the flowing contents' data. See 10.3.2 for details.

Example:

```
<body_module>
  <flow_type_body>
    <flow_entry>
      <flow_data ...> ... </flow_data>
      <flow_data ...> ... </flow_data>
      ...
    </flow_entry>
    <special_page_link>
      <special_page kind="contents">PG0001</special_page>
      <special_page kind="body">PG0002</special_page>
    </special_page_link>
    <search_table>
      ...
    </search_table>
  </flow_type_body>
</body_module>
```

10.3.2 Flowing content data <flow_type_body>

10.3.2.1 <flow_type_body> tag

The <flow_type_body> tag handles the flowing contents' data. It has the following child elements.

[Attribute]

None

[Child elements]

<flow_entry> Registers the flow data to be used as the main text's flowing content. There shall only be one instance of this element. Shall not be omitted. See 10.3.2.2 for details.

<special_page_link> Special page data. Allows specifying the position in the flowing content of some often needed pages, such as the index, or the beginning of the main content, for easy reference. May be omitted. Written as described in 10.3.2.5. Omission of this element means there is no special page.

<search_table> Records the data needed to create a search table. May be omitted. Written as described in 10.3.2.6. Omission of this element means there is no search table.

10.3.2.2 Flow data registering module <flow_entry>

The <flow_entry> tag registers the flow data to be used as the main text's flowing content. It has the following child elements.

[Child elements]

- <flow_default_attribute> Sets the default attributes to be used to display each flow data, as defined by the following <flow_data> element. May be omitted. See 10.3.2.3 for details. When omitted, the viewer should behave as if all of its attributes and child elements were set to their default value.
- <flow_data> Registers information on each flow data. There is a one to one relation between the number of <flow_data> tags and actual flow data to be recorded / displayed. There shall be one or more instances of this tag. The order in which flow data are recorded determines the display order. See 10.3.2.4 for details.

10.3.2.3 Flow data default attribute module <flow_default_attribute>

Sets the default values of attributes that will be used to display each flow data (see 10.3.2.4), of which the main flowing content is composed. The values set in this tag will serve as default values for all the flows which have a text object, a search screen object or dictionary data object as main data.

Part of the values that may be set in this tag may also be set locally in the *text object instance* of each content data. In order to set a default value for particular content data, the <text_default_attribute> of the *text object instances* (see 10.6.2) or <dict_default_attribute> of the dictionary data object designated by the content data should be used. If the value defined by the <flow_default_attribute> tag for the whole content data and the value set in an individual content data's <text_default_attribute> or <dict_default_attribute> are in conflict, the latter has priority. When neither is set, the behavior is not defined by this standard, and depends on the viewer's default, or the user's preferences. Moreover, if the viewer does not implement the required method, or wishes to give priority to user settings, it may proceed without respecting the value defined in these default value tags, except when the following explanations state otherwise.

The <flow_default_attribute> tag has the following attributes and child elements. If it is omitted, the viewer should behave as if all of its attributes and child elements were set to their own default value.

[Attributes]

- baseline: Defines the orientation of the baseline (and therefore of the text) for each flow. The possible values are listed below. May be omitted. If neither this attribute nor the corresponding option in each content data (the baseline attribute of the <text_default_attribute> tag in *text object instance* or <dict_default_attribute> in dictionary data object instance, see 10.6.2.2 and 10.6.3.2 for details) is set, the default value depends on the viewer.
 - "right": The writing direction is horizontal (left to right). However, the direction can be changed at the user's option.
 - "right_only": The writing direction is horizontal (left to right), and the user cannot change the setting. However, if it is not supported by the viewer, this setting is not necessarily applied.
 - "down": The writing direction is vertical (top to bottom). However, the direction can be changed at the user's option.
 - "down_only": The writing direction is vertical (top to bottom), and the user cannot change the setting. However, if it is not

supported by the viewer, this setting is not necessarily applied.

view_type: Defines the default screen orientation for each flow. The possible values are listed below. May be omitted. When omitted, the default value depends on the viewer.

"portrait": Chooses portrait (taller than wide) mode. However, it can be set to another direction at the user's option.

"portrait_only": Chooses portrait (taller than wide) mode, and the user cannot change the setting. However, if the viewer cannot handle this screen orientation, this is not necessarily applied.

"landscape": Chooses landscape (wider than tall) mode. However, it can be set to another direction at the user's option.

"landscape_only": Chooses landscape (wider than tall) mode, and the user cannot change the setting. However, if the viewer cannot handle this screen orientation, this is not necessarily applied.

lang: Defines the default language for each flow.

lang_code: Defines the default language code of the content. Defaults to "ISO 639-3". It is intended to specify a language that is not listed in the existing language code systems. It is recommended to conform to ISO 639-3 if possible. This is overridden by the lang_code attribute of tag.

"ISO 639-3": The lang attribute is written according to ISO 639-3.

Other values: Represent the language code system of the users's choice.

Example:

```
<flow_default_attribute base_line="right" view_type="portrait"
lang_code = "ISO 639-3" , lang = "jpn">
```

phonetic_notation: Defines the default pronunciation notation of the content. Defaults to "IPA". This is overridden by the phonetic notation attribute of <pronunciation> and <headword> tags inside such tags.

[Child elements]

<flow_default_size> Defines the default letter spacing, line pitch and margin size for all the flows of the content data. May be omitted. If it is, the viewer should behave as if all of its attributes were set to their default value. This tag has the following attributes.

[Attributes]

letter_spacing: Default letter spacing. The following values are allowed. May be omitted. If it is, the size is unspecified and depends on the viewer.

"maximum"

"big"

"medium"

"small"

"minimum"

	The actual sizes of these 5 possibilities are viewer-dependent, as it depends on the capabilities of the underlying device.
line_pitch:	Default line pitch. The following values are allowed. May be omitted. If it is, the size is unspecified and depends on the viewer. "maximum" "big" "medium" "small" "minimum" The actual sizes of these 5 possibilities are viewer-dependent, as they depend on the capabilities of the underlying device.
margin:	Default margin size. The following values are allowed. May be omitted. If it is, the size is unspecified and depends on the viewer. "big" "medium" "small" The actual sizes of these 3 possibilities are viewer-dependent, as they depend on the capabilities of the underlying device.
<flow_default_font>	Defines the default font name, size, and properties for all the flows of the content data. May be omitted. If an individual flow of the content data also defines it (with the <text_default_font> tag of the <i>text object instance</i>, as defined in 10.6.2.2 and <dict_default_font> tag of the dictionary data object instance as defined in 10.6.3.2, the individual values take precedence. If both are omitted, the behavior should correspond to the default value of the attributes listed below. The following attributes may be used.
[Attributes]	
fontname:	Default font name. More than one font may be specified. In that case, each font name should be separated by a comma (0x2c in Unicode). For instance: fontname="Aaa sans serif,Bbb gothic" The viewer should use the first listed font that is available. May be omitted. If both this attribute and the individual flow's attribute are omitted, the default value depends on the viewer.
fontsize:	Default font size. The following values are allowed. May be omitted. If it is, the size is unspecified and depends on the viewer. "maximum" "big" "medium" "small" "minimum"

The actual letter sizes corresponding to these 5 possibilities are viewer-dependent, as it depends on the capabilities of the underlying device.

bold_flag: Specifies if the content data should be displayed in bold or not. If omitted, the behavior depends on the viewer. The acceptable values are:

"yes": display as bold

"no": display normally

If set to "yes", all characters shall be displayed in bold style, except, in a *text object instance* or a dictionary data object instance, those within a tag with its bold_flag attribute set to "no" (see 10.6.2.3 and 10.6.3.4).

color_space, color, opacity:

Defines the font color to be used for the content data. Written in the standard *color* data format. If omitted, the value depends on the viewer.

ruby_flag: Defines whether ruby in the content data is to be viewed or not. When omitted, behavior depends on the viewer. The text to be displayed when ruby is turned on is the one included in the <ruby> tag of text object instances or dictionary data object instances. The following values may be used.

"yes": Ruby should be displayed, but can be turned off by the user.

"yes_only": Ruby shall be displayed, and cannot be turned off by the user. However, this does not apply to viewers not able to display ruby.

"no": It is recommended that ruby should not be displayed, but can still be turned on by the user.

"no_only": Ruby shall not be displayed, and cannot be turned on by the user. However, this does not apply if the viewer is incapable

italic: Decides if the content data should be displayed in italic or not. If omitted, the behavior depends on the viewer. The acceptable values are:

"yes": display in italic

"no": display normally

If set to "yes", all characters shall be displayed in italic style, except, in a *text object instance* or a dictionary data object instance, those within a tag with its italic attribute set to "no" (see 10.6.2.3 and 10.6.3.4)

oblique: Decides if the content data should be displayed in oblique or not. If omitted, the behavior depends on the viewer. The acceptable values are

"yes": display in oblique;

"no": display normally.

If set to "yes", all characters shall be displayed in oblique style, except, in a *text object instance* or a dictionary data object instance, those within a tag with its oblique attribute set to "no" (see 10.6.2.3 and 10.6.3.4)

small_caps: Decides if the content data should be displayed in small capitals or not. If omitted, the behavior depends on the viewer. The acceptable values are

"yes": display in small capitals;

"no": display normally.

If set to "yes", all characters shall be displayed in small capitals, except, in a *text object instance* or a dictionary data object instance, those within a tag with its small_caps attribute set to "no" (see 10.6.2.3 and 10.6.3.4)

family: Specifies font family to be used. It depends on the viewer what font is actually used in rendering. If omitted, the behavior is also viewer-dependent. The acceptable values are

"monospace": display in fixed-width fonts;

"sans-serif": display in fonts without serifs;

"serif" display in fonts with serifs;

"cursive" display in handwriting style fonts.

"fantasy" display in decorative fonts

If set to "yes", all characters shall be displayed in decorative style, except, in a *text object instance* or a dictionary data object instance, those within a tag with its fantasy_flag attribute set to "no" (see 10.6.2.3 and 10.6.3.4)

ul_type Specifies how strings in <u> tag and its equivalent () should be rendered. The acceptable values are as follows, all of which are taken from ISO/IEC 9541-1 (8.7.1.12.1.1 Score Name). Refer to this standard for the exact meaning of these values.

"rightscore"

"leftscore"

"throughscore"

em_type: Specifies how strings in tags should be rendered. The acceptable values are as follows, the first 4 of which are taken from ISO/IEC 9541-1: (8.7.1.12.1.1 Score Name). Refer to the document for the exact meaning of these values.

Note that when this attribute is specified, tag is no longer equivalent to .

"rightscore"

"leftscore"

"throughscore"

"kendot"

"bold": The substrings should be rendered in bold font.

"italic": The substrings should be rendered in italic font.

"bold_italic" The substrings should be rendered in bold italic font.

"reverse" The substring should be rendered with the characters and background reversed.

"shade" The substrings should be rendered with shade.

<flow_default_background> Defines the background color to be used for all the flows of the content data. If an individual flow of the content data also defines it (with the **<text_default_font>** tag of the *text object instance*, as defined in 10.6.2.2 or with the **<dict_default_font>** tag of the dictionary data object instance as defined in 10.6.3.2, the individual values take precedence. If both are omitted, the behavior should correspond to the default value of the attributes listed below. This tag has the following attributes.

[Attributes]

color_space, color, opacity:

Defines the font color to be used for the content data. Written in the standard *color* data format. If omitted, the value depends on the viewer.

<flow_default_line_breaking_method> Specifies the algorithm to be used to determine how text should be split in lines. Various languages handle this in various ways, so this element allows for some flexibility. May be omitted. If both this tag and **<line_breaking_method>** tags in the *text/dictionary data object instance* (defined in 10.6.2.2 and 10.6.3.2) are specified, the latter takes priority. If both are omitted, the default values of this **<flow_default_line_breaking_method>** tag define the behavior. This tag has the following attributes and child elements.

[Attributes]

method: Chooses the line breaking method. Defaults to "none". Possible values are:

"none": no special processing. When a line is filled with characters, go to the next one.

Other values are possible for localization. See Clause A.2.

Other attributes may be added by localized methods. See Clause A.2.

[Child elements]

None, unless added by localized methods. See Clause A.2

<flow_default_delimiter> Specifies the block for delimiters (character strings that appear before the first character and after the last character) for various elements in the content. Its child elements are tags that specify such delimiters for some tags, which serve as a guide when the content is converted into the data for the viewer, or rendered directly. **<flow_default_delimiter>** may be omitted, and all the delimiters default to "" (empty string) in such cases.

[Child elements]

<delimiter> Defines the starting and ending delimiters of the child element of the tag specified by the tag attribute.

It is also possible to specify the type attribute when applicable (i.e. the tag specified has the type attribute). The tag is to be ignored if the tag attribute specifies a non-text-rendering tag. When no delimiter is specified to a tag, it is regarded that an empty string "" is specified.

[Attributes]

tag: Specifies which tag the delimiters apply to. Ignored when a non-existing tag is specified. Shall not be omitted.

type: Specifies the value of the type attribute of the tag which the delimiters apply to.

When type attribute is specified to a tag that has no type attribute, the instance of <delimiter> tag is ignored. If multiple <delimiter> tags are given, and one of them has the type attribute while the others don't, the former overrides the latter when the former is applicable. May be omitted.

open: Defines the starting (opening) delimiter. May be omitted and defaults to "" (empty string).

close: Defines the ending (closing) delimiter. May be omitted and defaults to "" (empty string).

[Child elements]

None.

Example:

```
< flow_default_delimiter>
<delimiter tag="headword" type="pronunciation" open="/" close="/">
<delimiter tag="etymology" open="&lt;" close="&gt;">
<delimiter tag="p" open="*" close="*">
</ flow_default_delimiter>
```

10.3.2.4 Flow data <flow_data>

This is where flow data is defined. Under this tag, the object to be used as the flow's content is registered, as well as other information such as page links and events. This <flow_data> tag has the following attributes and child elements.

[Attributes]

flow_id: Sets the ID number of the flow data, written in the form of a *Page_ID*. May be omitted. Within <flow_entry>, there shall be no other flow data with the same ID.

body_id: Points to the object that will constitute this flow's content, using the *Object_ID* of a "flowing content text" object, or of a "search page" object, or a dictionary data object. Shall not be omitted.

turning_page_control: The viewer allows moving forward or backward in the contents. However, it is possible to restrict moves to the previous or next flow by setting this attribute to one of the values listed below. When omitted, it defaults to "off". Note, however, that when body id points to a search page object, turning_page_control shall be set to "on".

"on": Moving both to the next or previous flows is forbidden.

"off": Moving to the next and previous flows is permitted.

"forward": Moving to the next flow is forbidden, but moving to the previous one is allowed.

"back": Moving to the next flow is permitted, but moving to the previous one is forbidden.

As explained above, this setting limits moves from one flow to the others, but they do not restrict moves within each flow. To restrict moves inside a

flowing content text object, the turning page control of the <page_break> tag (see 10.6.2.3) shall be used in flowing content text object and of the <dic-item> tag in dictionary data object.

[Child elements]

<event_info> Event information module. Defines events (triggers) and associated actions. May be omitted. See 10.4 for details.

10.3.2.5 Special page data <special_page_link>

Records the position of important or frequently accessed pages, to make it easier to jump to these parts of the document. For instance, easy access to the map in a travel guide, to the chronology in a history book, or the glossary in a technical paper can prove useful. This <special_page_link> tag has the following child element.

[Child elements]

<special_page> Records the information about a single special page. It has the following attributes and child elements.

[Attributes]

kind: Describes the contents of the page referred to. The possible values are listed below. Defaults to "other" when omitted.

"cover", "title_page", "preface", "contents" (table of contents), "body" (beginning of the content), "column" (boxed piece of text), "note", "figure", "ad", "afterword", "appendix", "answer", "glossary", "bibliography", "commentary", "index", "imprint", "author_info", "other".

title: Defines a title for the position of the document referred to as a special page. Written as an *Extended character string*. May be omitted.

[Child elements]

Standard character string: Specifies the position to consider as a special page, by choosing a particular flow data from the whole flowing content. Shall not be omitted. For a textual flow data, it is possible to specify a position within it as well.

- To specify only the flow data as a whole.

Written as a *Page_ID*. It records the corresponding flow data's page ID.

Example:

```
<special_page ... >PG0001</special_page>
```

- To specify the flow data and the position within it.

Written as "*Page_ID/Object_ID/Char_ID*". Page ID is the flow data's ID number. Object ID is the ID number of a text/dictionary data object within that flow. Char ID is the ID number of a string defined within that text/dictionary data object.

Example:

```
<special_page ... >
    PG0001/OB0321/CR0982
</special_page>
```

Example:

```
<special_page_link>
    <special_page kind="contents">PG1111 </special_page>
    <special_page kind="other" title="Downtown area map">PG1114/OB0022/CR0001
    </special_page>
    ...
</special_page_link>
```

10.3.2.6 Search table data <search_table>

Information defining the search tables is stored here, such as the search table's ID, or parameters concerning entry words registered in the search table. Entry words hereafter means target of each search and represented in two different ways:

- registered in the <head> tag of the *text object instance* (see 10.6.2.3);
- registered in the <head> tag of the *dictionary data object instance* (see 10.6.3.4).

During the conversion to the distribution format, based on the parameter stored in the <search_table_def> tag defined below, the actual search table is built, after verification of the entry word strings. In the distribution format, entry words are stored in this search table. The <search_table> tag has the following attributes and child elements.

[Attributes]

bookmark specifies if all the search tables included in the content should be eligible for automatic bookmarking on the viewer (i.e. automatically keeping the tally of the headwords searched). The following values are possible. May be omitted and regarded as "no" when omitted.

"yes":	the search tables are eligible for automatic bookmarking.
"no":	the search tables are not eligible for automatic bookmarking.

wordbook specifies if all the search tables are eligible for the wordbook function on the viewer (i.e. automatically registering headwords in the wordbook and showing the list). The following values are possible. May be omitted and regarded as "no" when omitted.

"yes":	the search tables are eligible for wordbook function.
"no":	the search tables are not eligible for wordbook function.

jump_search_root specifies if the content is eligible as a source of multi-content search from text (i.e. can launch multi content search using the keyword selected from the displayed text of the content). The following values are possible. May be omitted and regarded as "no" when omitted.

"yes":	the content is eligible as a source of multi-content search from text.
--------	--

"no": the content is not eligible as a source of multi-content search from text.

`jump_search` specifies if all the search tables are eligible for multi-content search from text (i.e. can be searched as one of the contents in multi-content search using the keyword selected from the text) and multi-content search from input with the content being displayed (i.e. can be searched as one of the contents in multi-content search using the input keyword) with the content displayed. The following values are possible. May be omitted and regarded as "no" when committed.

"yes": the search tables are eligible for such searches.

"no": the search tables are not eligible for such searches.

`all_search` specifies if all the search tables are eligible for multi-content search without the content itself being displayed. The following values are possible. May be omitted and regarded as "no" when omitted.

"yes": the search tables are eligible for such searches.

"no": the search tables are not eligible for such searches.

If specified in the text object instances, only "no" is valid for `bookmark/wordbook/jump_search_root/jump_search/all_search` attributes.

[Child elements]

`<search_table_def>` Specifies information concerning each search table. There shall be at least one `<search_table_def>` tag in the `<search_table>` tag. The following attributes and child elements are available.

[Attributes]

`id`: Defines the ID number of this search table. The ID number shall be unique to each search table within the content and stored as a *Standard character string*. Shall not be omitted. During the conversion to the distribution format, the search table is built according to the parameters in this `<search_table_def>` tag, and added to the contents. This ID number is used to refer to the search table either from the search page's entry fields (see 10.6.7 for details) or from the entry word information stored in *text object instance's* `<head>` tag (see 10.6.2.3 for details). Not to be confused with Reference ID (see 10.6.3.4).

`use_default`: Defines whether a search based on this search table is allowed by the viewer, when the search is not initiated from a related search page. The following values may be used. Defaults to "no" when omitted.

"yes" such search is allowed.

"no" such search is not allowed.

`sorting_rule`: Defines how to sort the search results. This attribute is to allow for different ordering schemes for different languages. May be omitted. Defaults to "unicode". The following values may be used.

"unicode": The rank of each character is determined by its Unicode code point.

Other sorting methods are defined in the localization (see A.2.5).

- name:** Sets the name for the search table. Written in a *Standard character string*. The name is intended to be used in listing the search results. Shall not be omitted.
- short_name:** Sets the shortened version of the name of the table. Names longer than the one specified by the name attribute are not allowed. The name is intended to be used in listing the search results. Shall not be omitted.
- wild:** Specifies if the a search table for wildcard search (i.e. it is known how many letters need to be filled in the blanks of the word as well as the positions of the blanks) should be output. The following values are possible. Shall not be omitted.
- "yes" the table for wildcard search should be output.
 - "no" otherwise.
- blank:** Specifies if the a search table for blank word search (where only the positions of the blanks in the word are known) should be output. The following values are possible. Shall not be omitted.
- "yes" the table for blank word search should be output.
 - "no" otherwise.
- end:** Specifies if a search table for word ending search should be output. The following values are possible. Shall not be omitted.
- "yes" the table for word ending search should be output.
 - "no" otherwise.
- help_page_id:** Specifies the page ID for the flow data that are related to the table (e.g. explanation of the table). May be omitted.

[Child elements]

- <enable_key_type>** Defines the type of characters that may be used to store the lookup key of the entry words in the table. For some languages such as English, the lookup key will simply be the word itself. However, other languages such as Japanese may use the pronunciation of the word instead of its ideographic representation. Shall not be omitted. It has the following attributes.

[Attributes]

- numerals:** Defines whether the lookup key may include numerals or not. Possible values are "yes" and "no". Defaults to "no" when omitted.
- basic_alphabet:** Defines whether the lookup key may include alphabet (limited to the non-accentuated characters found in US-ASCII) or not. Possible values are "yes" and "no". Defaults to "no" when omitted.

Refer to A.2.6 on localization for language-specific attributes. The preceding character sets are defined in Table 5.

Table 5 – Characters usable for the lookup key

Character set name	Corresponding characters (all values are in Unicode)
numerals	numerals: 0 to 9 (0x0030 to 0x0039)
basic_alphabet	Standard US-ASCII alphabet: A to Z (0x0041 to 0x005A) and a to z (0x0061 to 0x007A)

<key_normalization> Defines the normalization methods to be applied on the keys registered in this search table. Shall not be omitted. The interpretation of this tag differs depending on the modes of the search, namely "matches-only" search and "matches-first" search. The mode of search to use is determined by the search_type attribute of the <key_input_region> tag (see 10.6.7 for details).

a) "matches-only" search

(Only the entries matching the input are displayed, being narrowed down as input is becoming complete.) Both user input and the keys registered in the search table are normalized according to the rules set by the following attribute, and matching is conducted on the normalized form.

b) "matches-first" search

(Entries are displayed starting from those matching the input.) The settings are ignored and the normalization is done according to the default values of the following attributes.

The following attributes are available.

[Attribute]

capitalization: Turn all (alphabetical) characters to upper case. Possible values are "yes" and "no". Defaults to "yes".

See the localization (see A.2.7) for language specific normalizations.

Example:

```
<search_table>
  <search_table_def id="ST0001" use_default="yes">
    <enable_key_type numerals="no" basic_alphabet="yes"/>
    <key_normalization capitalization="yes"/>
  </search_table_def>
  <search_table_def id="ST0002">
    ...
  </search_table_def>
</search_table>
```

10.4 Event info module <event_info>

10.4.1 Events

In this standard, audio playback to react to clicks, pages links, or other user activated functions, are called events. The event info module records the events of flow data. It has the following child element.

[Attribute]

display_type: Specifies environments for the events. Currently only "single" is possible. Defaults to "single" when omitted.

[Child elements]

<event> Event data. Records information concerning the events that occur after the object pointed by the **body_id** attribute of the **<flow_data>** tag has been displayed, such as user launched events. For instance, clicking on a string may initiate a jump to another page. See 10.4.2 for details. There may be more than one instances of this **<event>** tag within **<event_info>**. If no event is to be specified, the **<event_info>** tag itself shall be omitted.

Example:

```

<event_info display_type="single">
  <event>
    <trigger>
      <trigger_pointer id="OB0001/CR0001" action_flag="click"/>
    </trigger>
    <action>
      <action_page_jump page_id="PG0002"/>
    </action>
  </event>
  ...
  <event>
    <trigger>
      <trigger_pointer id="OB0001/CR0002" action_flag="click"/>
    </trigger>
    <action>
      <action_page_jump page_id="PG0003"/>
    </action>
  </event>
  ...
</event_info>

```

Event 1

Event2

10.4.2 Event data <event>

An event is composed of two parts: the trigger part, and the action part. The former is the condition that triggers the event while the latter describes what is to be done. For example, the trigger may be "the user clicks on a specific area", or "a button is pressed", and the action may be "jump to a specific page", or "play a given soundfile". This **<event>** tag has the following child elements.

[Child elements]

<trigger> See 10.4.3 for details. There shall be only one **<trigger>** tag, and it can have only one child element. Shall not be omitted.

<action> See 10.4.4 for details. There shall be only one **<action>** tag, and it can have only one child element. Shall not be omitted.

Example:

<event>	
<trigger>	
<trigger_pointer id=" OB0001/CR0002" action_flag="click"/>	
</trigger>	Trigger
<action>	
<action_page_jump page_id="PG0003"/>	
</action>	Action
</event_info>	

10.4.3 Trigger (pointer) <trigger_pointer>

The <trigger> tag has the following child elements.

Used to define triggers as click on an area. This <trigger_pointer> has the following child elements and attributes.

[Attributes]

id: points at the target of the trigger. Shall not be omitted. Written as *Object_ID/Char_ID*. Not to be confused with Reference ID (see 10.6.3.4).

Example:

id="OB003k/CR0023"

The char ID part of this string may only refer to the following (see 10.6.2.3 for details):

- the id number defined in the char_id attribute of an <char_id> tag;
- the id number defined in the char_id attribute of an <object> tag.

However, when a clickable image map is defined using the <pointer_region> tag, only the second case (char_id of an <object> tag) may be used.

Note that the id number set in the trigger attribute of a <mask> tag (see 10.6.2.3) shall not be used here. If used, the trigger attribute of the <mask> tag is ignored.

action_flag: The type of action which switches the trigger on. In the current standard, only "click" is allowed, and is used as a default value in case of omission.

"click": a click in the target zone.

[Child elements]

<pointer_region> When the trigger area is restricted to only a part of the image pointed to by the id attribute of the <trigger_pointer> tag, this <pointer_region> tag is used to describe a *Polygonal_region* (see 9.15). If what the id attribute of the <trigger_pointer> points to is not an image, the content of this element is ignored. May be omitted.

When events have overlapping triggers, there can be some ambiguity as to which event takes precedence when the overlapping areas are clicked, depending on what kind of pointing devices are used. In that case, events are given precedence in the order of appearance inside the <event_info>. The following example illustrates this situation.

Example:

```

<event_info>
  <event>
    <trigger>
      <trigger_pointer id="OB003k/CR0001" action_flag="click">
        <pointer_region>
          <vertex position="(0,0)"/>
          <vertex position="(100,0)"/>
          <vertex position="(100,100)"/>
          <vertex position="(0,100)"/>
        </pointer_region>
      </trigger_pointer>
    </trigger>
    <action>
      <action_page_jump page_id="PG0043"/>
    </action>
  </event>
  <event>
    <trigger>
      <trigger_pointer id="OB003k/CR0001" action_flag="click">
        <pointer_region>
          <vertex position="(50,50)"/>
          <vertex position="(100,50)"/>
          <vertex position="(100,100)"/>
          <vertex position="(50,100)"/>
        </pointer_region>
      </trigger_pointer>
    </trigger>
    <action>
      <action_page_jump page_id="PG0021"/>
    </action>
  </event>
</event_info>

```

In that case, both events are triggered by a click on a sub-area of OB003k/CR0001. These two areas overlap on the (50,50)-(100,100) square. A click in this particular area shall trigger the first of the two events, resulting in a page jump to PG0043.

10.4.4 Action information

10.4.4.1 <action> tag

The <action> tag has the following child elements. Only one of them may be used per <action> tag.

10.4.4.2 Playback action <action_play>

Launches the playback of a sound, and animation or other playable items. It has the following attributes.

[Attributes]

object_id: Points to the object to be played. Written as described below. Shall not be omitted.

If the object is registered in the object management table (see 10.5) (in the present version of this standard, only possible for sound and movie objects), then it is:

Written as an *Object_ID*.

Example:

```
object id="OB003k"
```

If the object is inserted in the *text/dictionary data object instance* by the `<object>` tag (in the present version of this standard, only possible for animations), then it is:

Written as *Object_ID/Char_ID*.

Example:

```
object id="OB003k/CR0023"
```

Note that the char ID part shall be an id number registered in the `char_id` attribute of an `<object>` tag (10.6.2.3).

action: Defines the playback method. The only accepted value is "normal", and it defaults to "normal" in case of omission. When the object to be played is a sound item, when the reader moves to another flow, or reaches a `<page_break>`, or `</word>` with `page_break` attribute set to "yes" (in dictionary data object instances only), the playback shall be stopped. In case of an animation, the playback stops when it goes out of the screen.

Example:

```
<action_play object_id="OBk123"/>
<action_play object_id="OB1234/CR0001"/>
<action_play object_id="OB1234/CR0001" action="normal"/>
```

10.4.4.3 Page jump action `<action_page_jump>`

Jumps from the current page to another one, or to a website. It has the following attributes.

[Attributes]

book: Defines the target document of the jump. If it is the same document as the origin, this attribute should be omitted. Otherwise, there are 3 different usages as listed below.

- a) Jump to a web site:
When the target is an html web site, the book attribute should be written as an URL address beginning with `http://` or `https://`, as defined in RFC 3986.
- b) Make a phone call:
The book attribute should be written as `"tel: "` followed by a telephone number.
- c) Write a mail to the specified address:
The book attribute should be written as `"mailto: "` followed by a single email address.

book_type: Used to differentiate the types of content stored in the book attribute. Shall be omitted if the book is omitted too, shall not be omitted when the book is not omitted.

- a) When the book attribute points to a web site.
The attribute shall contain: `"text/html"`

- b) When the book attribute points to a telephone number.

The attribute shall contain: "application/x-tel"

- c) When the book attribute points to a mail address.

The attribute shall contain: "application/x-mail"

page_id: Defines the target flow data of the jump, when the jump target and the jump origin are within the same document, and shall not be omitted in that case. Written as a *Page_ID*. If the book attribute is set to 1) website, 2) telephone, or 3) mail address, then this attribute is ignored.

center: Sets a more precise destination to the jump. Can only be used when the jump target is a text/dictionary data object (see 10.6.2 and 10.6.3). In that case, the position within that text is specified using the *char_id* of the target. May be omitted. When omitted, the jump target is the beginning of the flow data set by the *page_id* attribute. If the book attribute is set to 1) website, 2) telephone, or 3) mail address, then this attribute is ignored.

Example:

```
<action_page_jump page_id="PG23k4" />
<action_page_jump page_id="PG23k4" center="CR0001"/>
<action_page_jump book="http://www.sharp.co.jp/" book_type="text/html"/>
```

10.5 Parts data module <parts_module>

10.5.1 Storage and management

This subclause is used to store and manage information about the parts that are used to constitute the flow data. The <object_table> sets an object ID number and various attributes for all the objects that will be used as content, such as text objects.

<parts_module> has the following child element.

[Child elements]

<object_table> Object management module. All objects that are used in the book are registered here. Each object is recorded using one of the elements described in the following subclauses. Shall not be omitted.

[Child elements]

<dynamic_text_object_entry>	Records text objects serving as a base for the flowing contents.
<sound_object_entry>	Records sound objects.
<search_page_object_entry>	Records search page objects.
<movie_object_entry>	Records movie objects.
<dict_data_object_entry>	Records dictionary data objects.

Example:

```
<parts_module>
  <object_table>
    <dynamic_text_object_entry object_id="OBT001" ... >
      ...
    </dynamic_text_object_entry>
    <dynamic_text_object_entry object_id="OBT002" ... >
      ...
    </dynamic_text_object_entry>
    <sound_object_entry object_id="OBS001" ...>
      ...
    </sound_object_entry>
    <dynamic_text_object_entry object_id="OBT003" ... >
      ...
    </dynamic_text_object_entry>
    ...
    <search_page_object_entry object_id="OBSP01" ... >
      ...
    </search_page_object_entry>
    ...
    <movie_object_entry object_id="OBMV01" ... />
    ...
  </object_table>
</parts_module>
```

The following subclauses will explain how each object is registered.

10.5.2 Dynamic text object <dynamic_text_object_entry>

Records text objects to be used as part of the flowing content. This tag has the following child element and attributes.

[Attributes]

- src:** Filename of the *text object instance*. Shall not be omitted. Before opening the file specified in this attribute, it shall be checked against the type attribute.
- type:** Stores the MIME type of the *text object instance*. The only possible value is "text/x-bvf-text". Shall not be omitted.
- object_id:** Assigns an ID number to the object. This number is used from the flow data or event data modules to refer to this object. Written in the standard *object_ID* format. Shall not be omitted.

[Child elements]

- <permission_info>** Defines the permissions concerning the object pointed by the *src* attribute. Written in the standard *Permission* format. May be omitted.

Example:

```
<dynamic_text_object_entry src="sect1.xml" type="text/x-bvf-text" object_id="OB03k2"/>
<dynamic_text_object_entry src="sect1.xml" type="text/x-bvf-text" object_id="OB03k1">
  <permission_info>
    <print_permission permission="authorized"/>
  </permission_info>
</dynamic_text_object_entry>
```

10.5.3 Sound object <sound_object_entry>

Records sound objects. This tag has the following attributes and child element.

[Attributes]

type:	Records the MIME type of the sound object. In the current standard, the following types are permitted. Shall not be omitted.
MP3:	"audio/mp3"
AAC:	"audio/3gpp2"
SMAF:	"application/x-smaf"
src:	Filename of the sound object, written in the standard <i>Filename</i> format. Shall not be omitted. Before opening the file specified in this attribute, it shall be checked against the type attribute. Currently, only mp3 (extension: .mp3), AAC (extension: .3g2) and SMAF (extension: .mmf) are allowed.

[Child elements]

<sound_object_info>	Records information concerning the sound object. The following attribute is available.
---------------------	--

[Attribute]

object_id:	Assigns an ID number to the object. This number is used by the event data module to refer to this object. Written in the standard <i>object_ID</i> format. Shall not be omitted.
------------	--

Example:

```
<sound_object_entry src="foobar.mp3" type="audio/mp3">
  <sound_object_info object_id="OB143s"/>
</sound_object_entry>
```

10.5.4 Search page object <search_page_object_entry>

This records search page objects. It has the following attributes and child element.

[Attributes]

src:	Filename of the search page object, written in the standard <i>Filename</i> format. Shall not be omitted. Before opening the file specified in this attribute, it shall be checked against the type attribute.
type:	Records the MIME type of the search page object. In the current specification, only "text/x-bvf-search-page" is permitted. Shall not be omitted.
object_id:	Assigns an ID number to the object. This number is used by the event data module to refer to this object. Written in the standard <i>object_ID</i> format. Shall not be omitted.

[Child elements]

<permission_info>	Defines the permissions concerning the object pointed by the src attribute. Written in the standard <i>Permission</i> format. May be omitted.
-------------------	---

Example:

```
<search_page_object_entry
  src="spage1.xml" type="text/x-bvf-search-page" object_id="OBSP01" />

<search_page_object_entry
  src="spage2.xml" type="text/x-bvf-search-page" object_id="OBSP02" >
  <permission_info>
    <print_permission permission="authorized"/>
  </permission_info>
</search_page_object_entry>
```

10.5.5 Movie object <movie_object_entry>

This records movie objects. It has the following attributes and child element.

[Attributes]

- src:** Filename of the movie object, written in the standard *Filename* format. Shall not be omitted. Before opening the file specified in this attribute, it shall be checked against the type attribute. Currently, only 3GPP2 (extension: .3g2) is allowed.
- type:** Records the MIME type of the movie object. In the current standard, the following type is permitted. Shall not be omitted.
3GPP2: "video/3gpp2"
- object_id:** Assigns an ID number to the object. This number is used by the event data module to refer to this object. Written in the standard *object_ID* format. Shall not be omitted.

Example:

```
<movie_object_entry src="movie1.3g2" type="video/3gpp2" object_id="OBmv00"/>
```

10.5.6 Dictionary data object <dict_data_object_entry>

This records text dictionary data objects. This tag has the following child element and attributes.

[Attributes]

- src:** Filename of the *dictionary data object instance*. Shall not be omitted. Before opening the file specified in this attribute, it shall be checked against the type attribute.
- type:** Stores the MIME type of the *dictionary data object instance*. The only possible value is "text/x-bvf-dict-data". Shall not be omitted.
- object_id:** Assigns an ID number to the object. This number is used from the flow data or event data modules to refer to this object. Written in the standard *object_ID* format. Shall not be omitted.

[Child elements]

- <permission_info>** Defines the permissions concerning the object pointed by the *src* attribute. Written in the standard *Permission* format. May be omitted.

Example:

```
<parts_module>
  <object_table>
    <dict_data_object_entry src="dict1.xml" type="text/x-bvf-dict-data" object_id="OBD001">
      <permission_info>...</permission_info>
    </dict_data_object_entry>
    ...
  </object_table>
</parts_module>
```

10.6 Object instances

10.6.1 General

The term *object instance* is used in this standard to refer to the objects that are displayed (or played, as appropriate) by the viewer. The possible kinds of *object instances* are listed below:

Text object instance

Dictionary data object instance

Image object instance

Sound object instance

Animation object instance

Search page object instance

Movie object instance

10.6.2 Text object instance <text_data>

10.6.2.1 Text object instances

The *text object instances* are stored in XML files of their own, with <text_data> as the root element, as shown below.

Example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<text_data>
  <text_default_attribute>
    ...
  </text_default_attribute>
  <text_body>
    Two households, both alike in dignity, ...
  </text_body>
</text_data>
```

As can be seen, the <text_data> has two main child elements, one for storing the text's default attributes (<text_default_attribute>), and one to store the text itself (<text_body>).

10.6.2.2 Text default attributes <text_default_attribute>

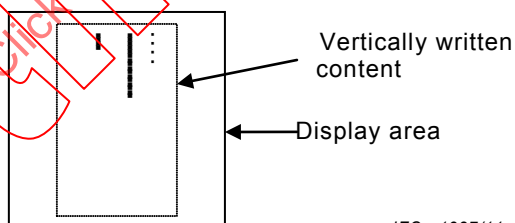
Text default attributes store parameters to use when displaying the text. Be aware that these parameters may not always be respected, if the viewer is not able to handle the values/behaviors set in this module, or if the viewer is configured to give precedence to user settings. In addition, as was explained in 10.3.2.3, some of the parameters defined here may also be set in the whole text flow (in the <flow_default_attribute> child element of

<flow_entry>. See 10.3.2.4). These are used as default if the local ones (the ones we define here) are not set. In case both are defined, the local settings take precedence.

The <text_default_attribute> tag has the following attributes and child elements.

[Attributes]

- baseline:** Defines the orientation of the baseline (and therefore of the text) for each flow. The possible values are listed below. Defaults to the <flow_default_attribute> value if omitted.
- "right": The writing direction is horizontal (left to right). However, the direction can be changed at the user's option.
 - "right_only": The writing direction is horizontal (left to right), and the user cannot change the setting. However, if it is not supported by the viewer, this setting is not necessarily applied.
 - "down": The writing direction is vertical (top to bottom). However, the direction can be changed at the user's option.
 - "down_only": The writing direction is vertical (top to bottom), and the user cannot change the setting. However, if it is not supported by the viewer, this setting is not necessarily applied.
- valign:** Determines how the text box is to be positioned. The vertical position for horizontally written text, or the horizontal position for vertically written text is what is determined here. The value described below is eligible. When omitted, the text box should be displayed from the top of the display area.
- "middle": The content of the <text_body> tag is centered within the display area, as shown below. If the content is larger than the display area, this attribute is ignored, see Figure 5.



IEC 1387/11

Figure 5 – Example of valign="middle"

[Child elements]

- <text_default_font>** Defines the default font and font color for this text object. When omitted, defaults to the <flow_default_attribute> value. It has the following attributes.

[Attributes]

- fontname:** Default font name. More than one font may be specified. In that case, each font name should be separated by a comma (0x2c in Unicode). For instance:
- fontname="Aaa sans serif,Bbb gothic"
- The viewer should use the first listed font that is available. When omitted, defaults to the <flow_default_attribute> value.
- color_space, color, opacity:**

Defines the font color to be used for the content data. Written in the standard *color* data format. When omitted, defaults to the <flow_default_attribute> value.

ruby_flag: Defines whether ruby in the content data is to be viewed or not. When omitted, defaults to the <flow_default_attribute> value. The text to be displayed when ruby is turned on is the one included in the <ruby> tag of text object instances. The following values may be used.

“yes”: Ruby should be displayed, but can be turned off by the user.

“yes_only”: Ruby shall be displayed, and cannot be turned off by the user. However, this does not apply to viewers not able to display ruby.

“no”: It is recommended that ruby not be displayed, but can still be turned on according to user preferences.

“no_only”: Ruby shall not be displayed, and cannot be turned off by the user. However, this does not apply if the viewer is incapable of disabling ruby display.

<text_default_background> Defines the background color and background image to be used for this text object. When both are defined, the image is drawn centered in the screen filled with the background color. Finally, the content of the <text_body> tag is rendered on the top. If the background image is too large to fit inside the screen, it is scaled down with the aspect ratio being preserved. The behavior in case of omission of this tag should conform to the behavior specified in case of omission of its attributes. The following attributes and child elements are available.

[Attributes]

color_space, color, opacity:

Defines the color to be used as background color. Written in the standard *color* data format. When omitted, defaults to the <flow_default_attribute> value.

type:

Stores the MIME type of the background image. May be omitted only if the src attribute is also omitted. The possible values are listed below.

"image/png"

"image/jpeg"

src:

Sets the filename of the image to use as a background image, written in the standard *Filename* format. May be omitted. Before opening the file specified in this attribute, it shall be checked against the type attribute.

[Child elements]

<permission_info>:

Defines the permissions about the image referred to by the src attribute. Written in the standard *Permission* format. May be omitted. If the src attribute is omitted, this permission information should be ignored.

<text_default_background_music>

Defines the background music to be played while displaying this text object. If this tag is omitted, nothing should be played. The following attributes and child elements are available.

[Attributes]

type:	Stores the MIME type of the background music. Shall not be omitted. The possible values are listed below. "audio/mp3" "audio/3gpp2" "application/x-smaf"
src:	Sets the filename of the background music, written in the standard <i>Filename</i> format. Shall not be omitted. Before opening the file specified in this attribute, it shall be checked against the type attribute.
loop:	Specifies whether the background music should be played iteratively (i.e. repeated from the beginning every time the end is reached). Possible values are "yes" and "no". In case of omission, no is assumed.

[Child elements]

<permission_info>:
Defines the permissions about the sound file referred to by the src attribute. Written in the standard *Permission* format. May be omitted.

<line_breaking_method> Specifies the algorithm to be used to determine how text should be split in lines. Various languages handle this in various ways, so this element allows for some flexibility. When omitted, defaults to the <flow_default_attribute> value. This tag has the following attributes and child elements.

[Attributes]

method:	Chooses the line breaking method. Defaults to "none". Possible values are: "none": no special processing, when a line is filled with characters, go to the next one. Other values are possible for localization. See Clause A.2. Other attributes may be added by localized methods. See Clause A.2.
---------	---

[Child elements]

None, unless added by localized methods. See Clause A.2.

Example:

```
<text_default_attribute>
  <text_default_font fontname="foobar-serif" color="green"/>
  <text_default_background color="black" src="001.jpg" type="image/jpeg"/>
  <text_default_background_music src="1.mp3" type="application/x-smaf"
loop="yes"/>
  <line_breaking_method ="none">
  </line_breaking_method >
</text_default_attribute>
```

10.6.2.3 Text body <text_body>

This part records the actual text body of the text object. In this tag, *Extended character string*, as well as the elements described below may be used.

a) Paragraph

<p>...</p> Creates a paragraph. Generally, there is a line break at the beginning of the paragraph (right before the <p> tag), and at its end (right after the </p> tag). However, there is no line break in the cases listed below.

If the paragraph starts at the beginning of a line even without inserting a line break.

If what follows the paragraph starts at the beginning of a line even without inserting a line break.

It should also be noted that should the paragraph end occur in text flowing around a image (see the align attribute of the <object> tag), the line break does not cancel this effect. Therefore, the next line still flows around the image. In order to cancel the flowing, and start the new line after the image, <br clear="all"/> shall be explicitly used. <p> has the following attributes and child elements.

[Attributes]

top_line_indent: Sets the size (expressed in em) of the indentation of the first line of the paragraph. The spaces are inserted before the first character following the <p> tag, and the first character following a
 tag in the paragraph. The unit is "em". The actual spacing is the sum of the value set in this attribute and the margin of the paragraph set by the "top" attribute. Moreover, this attribute may be set to a negative value, like "-2em". When omitted, defaults to 0.

top: Defines the size of the left margin of the paragraph (top margin in case the writing direction is vertical). It is expressed either in em or in a percentage of the line length (column in case the writing direction is vertical). When omitted, defaults to "0em". Negative values shall not be used. Paragraphs may be contained in another paragraph. In that case, the margins are added. Percentages are also relative to the inner area thus calculated, not the total display area.

bottom: Defines the size of the right margin of the paragraph (bottom margin in case the writing direction is vertical). It is expressed either in em or in a percentage of the total line size (column in case the writing direction is vertical). When omitted, defaults to "0em". Negative values shall not be used. Paragraphs may be contained in another

paragraph. In that case, the margins are added. Percentages are also relative to the inner area thus calculated, not the total display area.

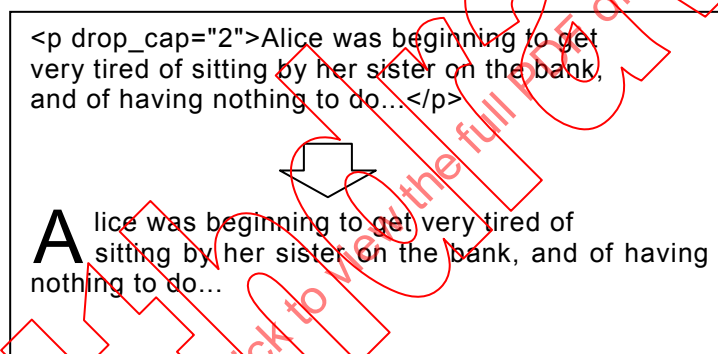
align: Determines whether the text should be in the center, left-aligned or right-aligned within the line size defined by the top, bottom and first line indent attributes. This attribute accepts the values listed below. When omitted, the current setting is kept unchanged.

"center": The string included in the <p> tag is displayed in the center.

"right": The string included in the <p> tag is right-aligned, or bottom-aligned in the case of vertical writing.

"left": The string included in the <p> tag is left-aligned, or top-aligned in the case of vertical writing.

drop_cap: Allows to turn the first letter of the paragraph into a dropped capital. The value is an integer representing the number of lines that the dropped capital should cover. Defaults to "1" (Normal size), see Figure 6.



IEC 1388/11

Figure 6 – Example of dropped capital

[Child elements]

Except the <page_break/> and <head> tags, any element that may be inside <text_body>, as well as *Extended character strings* may be used as the content of the <p> tag.

Figure 7 shows an example of a horizontally written paragraph with top="25%" and bottom="2em".

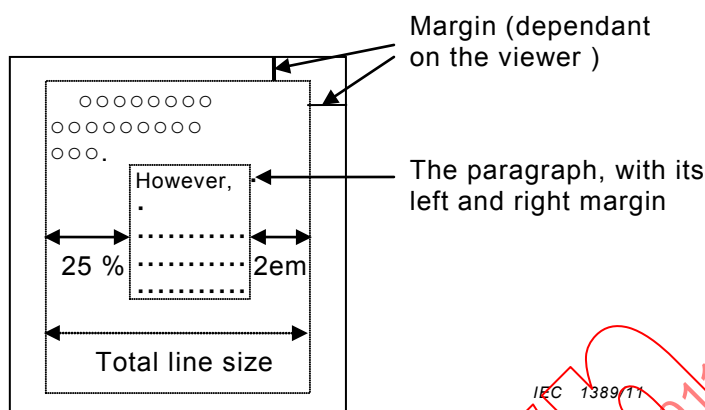


Figure 7 – Left and right margin of a paragraph

Example:

```
<p>.... </p>
<p align="center"> ... </p>
<p top="25%" bottom="2em"> ... </p>
```

<scrolling_text> Defines a line of scrolling text. Line breaks are automatically inserted around this element (right before the **<scrolling_text>** tag and right after the **</scrolling_text>** tag), to guarantee that the scrolling text is alone on its line. The height of the scrolling text object depends entirely on its content. In the case of horizontally written text, scrolling is done right to left, while for vertical text it should move the text from the bottom, upwards. The text should be scrolled until it disappears from the display area (limited by the "top" and "bottom" margin of the surrounding paragraph if there is one), and then start again. The **<scrolling_text>** tag accepts the following child elements.

[Child elements]

The string to be displayed as scrolling text is written as a child element, in the form of an *Extended character string*. The following tags may also be used: **<external_char>**, ****, **<horizontal>**, **<ruby>**, **<sub>**, **<sup>** and **<object>**. However, the **<object>** tag has the following additional limitations when used as a child element of the **<scrolling_text>** tag:

- only image files may be designated by the src attribute;
- the align attribute may only be set to "top", "middle", or "bottom";
- the char_id attribute shall not be used.

Example:

```
<scrolling_text>
  This text will be scrolling over and over
</scrolling_text>
```

b) Inline elements

**
** Inserts a line break. This tag shall not have child elements.

[Attribute]

clear: Cancels the text's flowing around an embedded object, such as an image. The
 tag does not do so when this attribute is omitted. The following values are possible.

"left": Inserts blank lines after the
 tag until the text can be inserted at the beginning of the line (left for horizontal writing, top for vertical). In other words, it moves the insertion point after the object, if any, around which the text flows, if it is on the left.

"right": Inserts blank lines after the
 tag until the text can be inserted until the end of the line (right for horizontal writing, bottom for vertical). In other words, it moves the insertion point after the object, if any, around which the text flows, if it is on the right.

"all": Inserts blank lines after the
 tag until the text can be inserted from the beginning to the end of the line. In other words, it moves the insertion point after the object, if any, around which the text flows, regardless of its being on the left or right.

Example:

```
<br/>
<br clear="all"/>
```

<hr/>

Draws a horizontal line (a vertical one when the text is written vertically). As this line shall always be alone in its row (column when the text is vertical), a line break (equivalent to
) is always inserted after it, and before it also, unless the line starts from the top on its own. It has the following attributes, but no child element.

[Attributes]

size: Determines the line's thickness. Expressed in "em". Note that this value does not need to be an integer. When omitted, it defaults to "0,1em".

length: Determines the line's length. Expressed in "em" or as a percentage of the available space (row if the text is written horizontally, column if written vertically), taking into account the enclosing <p> tag's "top" and "bottom" attributes. When omitted, it defaults to "100%".

align: Sets the position of the line if it is drawn shorter (using the length attribute) than the available area. Possible values are "left", "center" and "right". Defaults to center when omitted.

Example:

```
<hr/>
<hr size="0.5em" length="50%" align="center"/>
```

... Defines font properties for the text enclosed in the tag. It has the following attributes and child elements.

[Attributes]

name: Defines the font to use, by specifying the font name. If omitted, the font used before the **** tag will continue to apply. More than one font names may be specified. In that case, each font name should be separated by a comma (0x2c in Unicode). For instance:

fontname="Aaa sans serif,Bbb gothic"

The viewer should use the first listed font that is available.

size: Sets a different font size. The size is set as a percentage of what is set by the base attribute. If omitted, the base size is not modified.

base: Defines the base of calculation for the size attribute. The values listed below are accepted. If omitted, it defaults to "last"

"last": Same size as the character before the **** tag.

"default": Same size as the default font size.

color_space, color, opacity: Changes the character's color. Written in the standard *Color* data format (see 9.16 for details). If omitted, the color used for the characters immediately before the **** tag are used.

bold: Defines whether the string included in the **** tag should be displayed in bold face or not. The acceptable values are:

"yes": display in bold

"no": display normally

When omitted, the current state is kept.

underline: Defines whether the string included in the **** tag should be underlined or not.

"yes": display with underline

"no": display normally

It should be noted that objects (e.g. image objects) included inside the **** tag should not be underlined even if this attribute is set to "yes". When omitted, the current state is kept.

italic: Decides if the content data should be displayed in italic or not. If omitted, the behavior depends on the viewer. The acceptable values are:

"yes": display in italic

"no": display normally

When omitted, the current state is kept.

oblique: Decides if the content data should be displayed in oblique or not. If omitted, the behavior depends on the viewer. The acceptable values are:

"yes": display in oblique

"no": display normally

When omitted, the current state is kept.

small_caps: Decides if the content data should be displayed in small capitals or not. If omitted, the behavior depends on the viewer. The acceptable values are:

"yes": display in small capitals

"no": display normally

When omitted, the current state is kept.

family: Specifies the font family to be used. It depends on the viewer what font is actually used in rendering. If omitted, the behavior is also viewer-dependent. The acceptable values are:

"monospace": display in fixed-width fonts

"sans-serif": display in fonts without serifs

"serif" display in fonts with serifs

"cursive" display in handwriting-style fonts

"fantasy" display in decorative fonts

When omitted, the current state is kept.

ul_type Specifies how strings in <u> tag and its equivalent rendered. The acceptable values are as follows, all of which are taken from ISO/IEC 9541-1 (8.7.1.12.1.1 Score Name). Refer to this standard for the exact meaning of these values.

"rightscore"

"leftscore"

"throughscore"

em_type: Specifies how strings in tag and its equivalent rendered. The acceptable values are as follows, the first 4 of which are taken from ISO/IEC 9541-1:(8.7.1.12.1.1 Score Name). Refer to the document for the exact meaning of these values.

Note that when this attribute is specified, tag is no longer equivalent to

"rightscore"

"leftscore"

"throughscore"

"kendot"

bold: The substrings should be rendered in bold font.

italic: The substrings should be rendered in italicfont.

bold_italic: The substrings should be rendered in bold italic font.

"reverse" The substring should be rendered with the characters and background reversed.

"shade" The substrings should be rendered with shade.

[Child elements]

Any combination of *Extended character strings* and all the inline elements (except <page_break/>) and <object> may be used.

Example:

```
<font name="Aaa sans serif">.....</font>
<font size="200%">.....</font>
```

- <i>...</i>** Specifies the substrings should be displayed in italic. Equivalent to `<font italic="yes">...</font>`
- <u>...</u>** Specifies the substrings should be displayed being underline. Equivalent to `<font underline="yes">...</font>`. It has the following attribute.
- ul_type** Specifies how strings in `<u>` tag and its equivalent should be rendered. The acceptable values are as follows, all of which are taken from ISO/IEC 9541-1 (8.7.1.12.1.1 Score Name). Refer to this standard for the exact meaning of these values.
- The acceptable values are taken from ISO/IEC 9541-1 and are as follows:
- "rightscore"
- "leftscore"
- "throughscore"
- ...** Specifies the substrings should be displayed in bold italic. Equivalent to `<font bold="yes" italic="yes">...</font>` in default settings. If `em_type` attribute in `<flow_default_font>` and `<text_default_font>` is specified, this is not the case and it depends on the `em_type` attribute how the strings in this tag should be rendered. The acceptable values are as follows, the first 4 of which are taken from ISO/IEC 9541-1 (8.7.1.12.1.1 Score Name). Refer to the document for the exact meaning of these values.
- em_type:** Specifies how strings in `<em>` tag and its equivalent are rendered. May be omitted. When specified, this overrides what was specified by the `em_type` attribute of `<flow_default_font>`. The acceptable values are as follows, the first 4 of which are taken from ISO/IEC 9541-1.
- "rightscore"
- "leftscore"
- "throughscore"
- "kendot"
- "bold": The substrings should be rendered in bold font.
- "italic": The substrings should be rendered in italic font.
- "bold_italic": The substrings should be rendered in bold italic font.
- "reverse" The substring should be rendered with the characters and background reversed.
- "shade" The substrings should be rendered with shade.
- <oblq>...</oblq>** Specifies the substring should be displayed in oblique. Equivalent to `<font oblique="yes">...</font>`
- <sc>...</sc>** Specifies the substring should be displayed in small capitals. Equivalent to `<font small_caps="yes">...</font>`
- <mlg>...</mlg>** Represents a multiline note. Although this standard does not specify how multiline notes should be rendered, two examples of how they are to be rendered are shown below for a better understanding of the concept.

Example:

これが割注<mlg>わりちゅう</mlg>です。



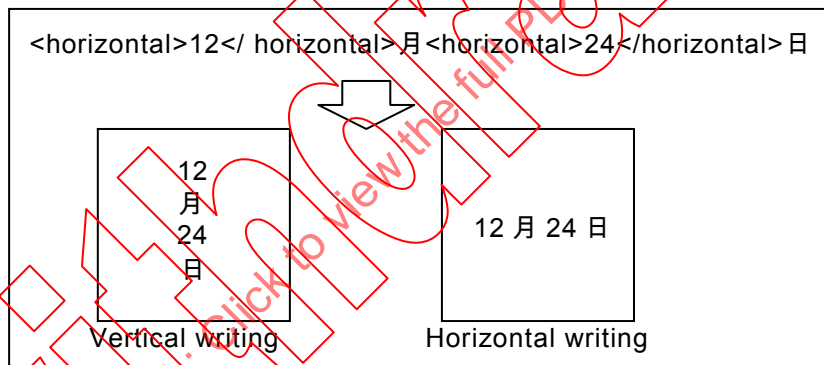
これが割注^{わりちゅう}です。

これが割注 (わりちゅう)です。

<horizontal>...</horizontal> Defines substrings that should be displayed horizontally even when the general text direction is vertical. This is often used for dates and other numbers in Japanese texts, see Figure 8. When the text is written horizontally, this tag has no effect. The following child element is accepted.

[Child elements]

External extended character strings may be used here.



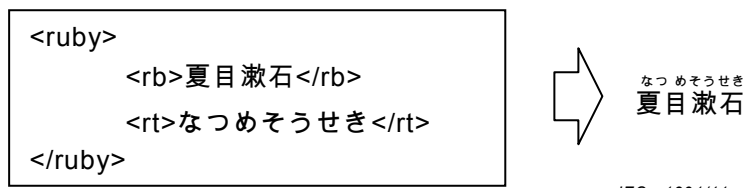
IEC 1390/11

Figure 8 – Horizontal writing in vertical text

<ruby>...</ruby> Used to display ruby text. Ruby is an often used feature in Asian, and particularly Japanese typography. It consists in a short run of text display above the base text, mostly used as a pronunciation guide for ideographic characters. As far as the available fonts permit it, ruby text is usually displayed in smaller characters than the base text it annotates. How line spacing is handled to open enough space to display ruby depends on the viewer. It should also be noted that as ruby is fundamentally a reading aid, and some people may not want or need it, some viewers may have an option to disable ruby displaying altogether, see Figure 9. The <ruby> tag has the following child elements.

[Child elements]

- <rb> The base string which should be annotated by a ruby. Shall not be omitted. Consists of an *External extended character string*.
- <rt> The ruby string that will be displayed above the base (or on the right side, if displaying vertical text). Shall not be omitted. Consists of an *External extended character string*.



IEC 1391/11

Figure 9 – Ruby

`<sub>...</sub>` Represents the substring as subscripts.

`<sup>...</sup>` Represents the substring as superscripts.

`<external_char>` Inserts an *External character*. The detailed syntax is described in 9.11. Within the *text object instance*, the `alt_img` and `alt_vimg` attributes may be used.

Example:

```
<external_char alt_set="oooextchars" alt_code="0x1234"
alt_img="ou.img" alt_vimg="ou_v.img"
img_type="image/jpeg" alt="鷗"/>
```

`<mask>` Adds a masking capability to the string embedded in this tag. When this string or one designated by the trigger attribute is clicked, the child element string is covered and uncovered alternately. It has the following attributes and child elements.

[Attributes]

`initial_flag`: Determines the initial state of the mask. The possible values are listed below, with "on" being the default in case of omission.

"on": The content is hidden by the mask.

"off": The content is not hidden.

`trigger`: Designates the string(s) that trigger the mask's status change. When a string indicated by this attribute is clicked, the mask should toggle between on and off. Written as one or more *Char_IDs* with comma signs (0x2C in Unicode) separating them in the latter case, as seen in the following.

Example:

```
"CR0001,CR0002,CR0003"
```

The designated character string shall be in the same *text object instance*. Note that the char id used here shall not be specified at the same time as a trigger in the event info module. If it is specified, no action is triggered on the mask even if the designated string is clicked. When omitted, the string enclosed in the mask itself serves as a trigger.

- char_id:** Sets a string ID to this <mask> object. Written in the standard *Char_ID* format. May be omitted.
- mask_type:** Selects the type / style of mask to be used. The values listed below may be used. Defaults to "default" when omitted.
- "default": The appearance of the mask depends on the viewer.
 - "color": The mask is a filled, colored box. The color is either defined by the corresponding attributes, or their default value in case of omission.
- color_space, color, opacity:** Defines the color of the mask, using the standard *Color* format, as described in 9.16. Shall be defined only if the mask_type attribute is set to "color". If omitted, defaults to "black".
- hold_flag:** Defines the state recording policy, allowing remembering whether the mask has been turned on or off. May be set to the values listed below. Defaults to "scope" when omitted.
- "scope": Remembers the state as long as the current text object instance is displayed.
 - "on_power": Remembers the state as long as the document is open.
 - "save": The state should be saved when closing the document and restored next time it is opened.

[Child elements]

The content of this element may be any combination of *External extended character strings* and the
, , <horizontal>, <ruby> and <object> tags. However, the following restrictions apply to the <object> tag:

The src attribute shall point to an image file.

The char_id attribute shall not be used.

Example.

The answer is <mask>42 </mask>.
The answer is <mask trigger ="CR0002">42</mask>.

<char_id> </char_id> Attaches a string ID number to its content. It has the following attribute and child elements.

[Attribute]

char_id: Sets the ID number of the tag's content, written in the *Char_ID* standard format. Shall not be omitted.

[Child elements]

The content of this element may be any combination of *Extended character strings*, inline elements except <page_break>, and the <object> tag.

<head> </head> Defines a list of keywords to be used for searching capabilities. The following child elements are available. It is possible to write string right after

the `</head>` tag. The interpretation for such a description is left to the viewer and the conversion (if any).

[Child elements]

`<headword>` Records a single keyword. There shall be at least one `<headword>` per `<head>`. It has the following attributes and child elements. It can also be used for registering the word for inclusion in a search table by specifying `table_id` attribute.

[Attributes]

`type`: specifies the type of the word in a *Standard character string*. The following values are pre-defined while other user-defined values are allowed as well. However, it is strongly recommended that "pronunciation" be used when applicable (i.e. the string represents phonetic symbols).

"pronunciation": The string represents pronunciation symbols.

other values: Other user-defined values are possible.

`table_id`: Points to the search table to which this keyword is to be added, by using the table's ID number, which is defined in `<search_table_def>` (see 10.3.2.6). Shall not be omitted if the content has a search page object (see 10.6.7). Not to be confused with Reference ID (see 10.6.3.4).

`phonetic_notation`: Denotes the phonetic alphabet used. Defaults to "IPA". To be ignored when the type is not "pronunciation". The following values are pre-defined while other user-defined values are allowed as well. However, it is strongly recommended that "IPA" be used when reasonably applicable.

"IPA": Denotes "International Phonetic Alphabet",

[Child elements]

Inside the tag is stored the text that should be displayed in the search results when the keyword matches. It is simply an *External extended character string*.

`<key> </key>` Defines the actual keyword. Only the characters allowed in the corresponding search table may be used if `<search_table>` exists. See `<enable_key_type>` for details in 10.3.2.6. May be omitted.

[Attribute]

`type`: Specifies the type of the keyword in *Standard character string*. May be omitted.

Example:

```
<head>
  <headword table_id="ST0001">color</headword>
    <key>color</key>
    <key>colour</key>
  <headword table_id="ST0002">color,colour</headword>
    <key>カラー</key>
</head>
<meaning> Visual effect caused by the wavelength of light...
```

<meaning> </meaning> Gives definitions.

[Attributes]

type: Specifies the category of the data. There are no predefined categories.

subid: Specifies a Reference ID to refer to a position in the definition.

level: Specifies the hierarchical level of the meaning in numerals [0-9]. Recorded in the order of importance (i.e. the smaller the nearer to the top of the hierarchy) though it is not required that the values be consecutively used.

no: Specifies the number of the meaning in the associated level in numerics [0-9].

<example> </example> Gives examples. When an example is given in bilingual format, it can either be represented using tags with the lang attribute set to corresponding languages, or as a plain text with a delimiter implicitly predefined as appropriate. (See example below.)

Example:

```
<example><span lang="eng" lang_code="ISO 639-3">I'll take
eel.</span><span lang="jpn" lang_code="ISO 639-3">僕はうなぎだ。
</span></example>
<example>I'll take eel|僕はうなぎだ。 </example>
```

<subhead> </subhead> Defines sub-headwords and related data. Intended to be used for introducing derived and compound words, etc.

[Attributes]

subid: Defines a Reference ID for this entry. May be omitted.

type: Defines type for the subhead word (e.g. derived word, compound word) in a *Standard character string*. May be omitted.

[Child elements]

<subheadword> </subheadword> Defines sub-headwords.

[Attributes]

subid: Defines a Reference ID for the subhead word. May be omitted.

type: Defines type for the subhead word (e.g. derived word, compound word) in a standard character string. The following values are pre-defined while other user-defined values are allowed as well. However, it is strongly recommended that "pronunciation" be used when applicable. May be omitted.

"pronunciation": The string represents pronunciation symbols.

[Child elements]

Inside the tag is stored the text that should be displayed in the search results when the keyword matches. It is simply an *External extended character string*.

<key> </key> Defines the actual keyword. Only the characters allowed in the corresponding search table may be used if <search_table> exists. See <enable_key_type> for details in 10.3.2.6. May be omitted.

[Attribute]

type: Specifies the type of the keyword in a *Standard character string*. May be omitted.

<meaning> See the explanation of the <meaning> tag above for details.

<example> Gives definitions. See the explanation of the <example> tag above for details.

<split> </split> Specifies the delimiter between lexicographical blocks (e.g. this tag may be used between the last word starting with "A" and the first word starting with "B"). This tag is intended to record the starting points of such blocks as well as what strings should be rendered for that (e.g. "-B-" for a block consisting of words starting with "B".)

[Attribute]

level: Defines hierarchical levels in numerals [0-9]. Recorded in the order of hierarchy (i.e. the smaller, the nearer to the top of the hierarchy) though it is not required that the values be consecutively used.)

[Child elements]

The same as <p>.

Example

```
<dict_body>
<split>- A -</split>
<dic-item> ... </dic-item>
<dic-item> ... </dic-item>
:
<dic-item>...</dic-item>
<split>- B -</split>
<dic-item>...</dic-item>
:
<dict_body>
```

<column> **</column>** Introduces articles.

[Attributes]

subid: Defines a Reference ID for the column.
type: Defines the type of the column in a *Standard character string*.

[Child elements]

The same as **<p>**.

<page_break/> Inserts a page break. This tag has attributes listed below. It shall not be used as a child element of any of the tags as that may be used inside **<text_body>**. Note that when **<page_break/>** is the last element of **<text_body>**, a blank page is inserted.

[Attribute]

turning_page_control: Defines how ordinary scrolling interacts with the page breaks, by allowing or forbidding crossing the page break forward or backward. The possibilities are listed below.

"on": The page break shall not be crossed by scrolling.

"off": The page break can be crossed by scrolling.

"forward": The page break can be crossed by scrolling only when moving forward.

"back": The page break can be crossed by scrolling only when moving backward.

When omitted, the behavior should conform to the global setting defined for the *text object instance* (the **turning_page_control** attribute of **<flow_data>**, see 10.3.2.4).

Example:

```
<page_break/>
<page_break turning_page_control="off"/>
```

c) Object

`<object>...</object>` Inserts an external object in the text body. It has the following attributes.

[Attributes]

type: Specifies the type of the inserted object, by giving the MIME type of the file. Shall not be omitted. In the current version of this standard, the following types are permitted:

Png image: "image/png"

Jpeg image: "image/jpeg"

Mp3 audio data: "audio/mp3"

Animation object: "application/x-bvf-flip-animation"
(see 10.6.6)

src: Sets the filename of the object to be inserted, written in the standard *Filename* format. Shall not be omitted. It shall be checked against the type attribute before the file referred to by this src attribute is read.

char_id: Assigns a string ID number to this object, written as a *Char_ID*. May be omitted.

align: Defines how text should flow around the inserted object. The possible values are listed below. When omitted, it defaults to bottom for horizontally written text, and to middle for vertically written text.

"top": Inline display. When text is written horizontally, the topmost part of the row should be aligned with the upper edge of the object; when written vertically, the rightmost part of the column should be aligned with the right edge of the object.

ABCDEFGH  IJKLMNOP

"middle": Inline display. When text is written horizontally, the object should be vertically centered on the row's base line. When writing vertically, the object should be horizontally centered on the column's base line.

ABCDEFGH  IJKLMNOP

"bottom": Inline display. When text is written horizontally, the base line of the row should be aligned with the lower edge of the object; when written vertically, the base line of the column should be aligned with the left edge of the object.

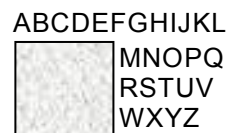


"left":

Flow around. When displaying horizontal text, the object is displayed next to the left margin, and the text is written in the space between the object and the right margin. When writing vertically, the object is displayed at the top, and text below it.

If the object is larger than the space for one row/column, there may be more than one rows/columns of text beside it. The align attribute of the surrounding paragraph (the <p> tag) does not influence how the object is positioned. Even if set to something different from "left", the object is displayed along the left margin. The paragraph settings, however, still apply to the text flowing around the object. When the <object> tag appears on the middle of a text row, it is displayed at the beginning of the next line.

The following example should be rendered the same way with the <object> tag at any position between after the letter "A" and before the letter "M".



"right":

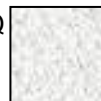
Flow around. When displaying horizontal text, the object is displayed next to the right margin, and the text is written in the space between the object and the left margin. When writing vertically, the object is displayed at the bottom, and the text is above it. If the object is larger than the space for one row/column, there may be more than one row/column of text beside it.

The align attribute of the surrounding paragraph (the <p> tag) does not influence how the object is positioned. Even if set to something different from "right", the object is displayed along the right margin. The paragraph settings, however, still apply to the text flowing around the object. When the <object> tag appears on the middle of a text row, it is displayed on the right of the next line.

The following example should be rendered the same way with the <object>

tag at any position between after the letter "A" and before the letter "M".

ABCDEFGHIJKL
MNOPQ
RSTUV
WXYZ



start: When the src attribute points to an animation object, this allows to define when the playback is started. Can only be used with animation objects. The possible values are listed below. When omitted, defaults to "auto". Moreover, when set to auto, the animation shall not be used by the <action_play> (see 10.4.4.2) tag.

"auto": Playback starts automatically, as soon as the object is displayed, running the animation from the start. If the animation is hidden and displayed again (because of scrolling), it should start again from the beginning, regardless of where it stopped.

"event": Playback is handled according to event data (see 10.4.2).

loop: Sets the number of times an animation should be played. In the present version of this document, the only accepted value is "1". When omitted, the animation object is played infinitely looping, until it goes out of the display area. However, if the start attribute is not set to "auto", the value of this loop attribute is ignored, and the behavior should be as if it were set to "1".

[Child elements]

<permission_info> Defines the permissions about the object referred to by the src attribute. Written in the standard *Permission* format. May be omitted.

Example:

```
<object type="image/png" src="image1.png"/>
<object type="image/jpeg" src="image2.jpg">
  <permission_info>
    <print_permission permission="authorized"/>
  </permission_info>
</object>
<object type="application/x-bvf-flip-animation"
  src="anime.xml" loop="1"/>
```

d) Logical block elements

<div>...</div> Represents a logical block. May be nested.

[Attributes]

level: Records the depth of the level in numerals [0-9]. May be omitted. Recorded in the order of importance (i.e. the smaller, the nearer to the top of the hierarchy) though it is not required that the values be consecutively used.

title: Records the title for the block. May be omitted. Intended to record the information.

[Child elements]

The same as <p>.

... A tag to be used to distinguish substrings that are not to be reflected in the final rendering, such as editor's memorandum. It is to be ignored by the viewer. This is also intended to be used as delimiters in <example> tags.

[Attributes]

type: Specifies the type of the substring. In *Standard character string*. May be omitted.

lang: Specifies in which language the substrings are written. May be omitted.

lang_code Specifies what language code system is employed. In the current specification, only "ISO 639-3" is predefined. Defaults to "ISO 639-3". May be omitted.

[Child elements]

The same as <p>.

Example:

```
<example><span lang="eng" lang_code="ISO 639-3">I'll take eel.</span><span lang="jpn" lang_code="ISO 639-3">僕はうなぎだ。</span></example>
```

e) Table elements

<table>...</table> Introduces html-like table. This is meant to arrange data into rows and columns on rendering, and it is not to be confused with search tables (see 10.3.2.6).

[Child elements]

<tr>...</tr> Defines rows in the table. <table> needs to have at least one instance of the <tr> tag as a child element.

[Child elements]

<td>...</td> Defines cells in the row. <tr> needs to have at least one instance of the <td> tag as a child element.

<th>...</th> Defines a header for the table. May be omitted.

f) Display according to the environment

<select>...</select> Represents alternative strings to be displayed according to the environment.

[Child elements]

<select_item> Defines strings for each environment.

[Attributes]

default:	Specifies that the instance of the tag gives the default environment when this attribute is set to "yes." Otherwise it is ignored.
type:	Specifies the type of the environment (e.g. the name of the hardware for which the content is intended.) in a <i>Standard character string</i> . Shall not be omitted.

g) TTS (text-to-speech) functionality

<tts>...</tts> Specifies the substring to be fed into text to speech representation.

[Attribute]

speaker:	Specifies the type of the speaker. Accepted values are:
"male":	grown-up male
"female"	grown-up female
"child"	child

h) Event

<a> Specifies an anchor to the link in a similar manner to the corresponding <a> element of HTML.

[Attribute]

href. Specifies the location of a Web resource.

10.6.3 Dictionary data object instance**10.6.3.1 General**

The dictionary data object instances are stored in XML files of their own, with <dict_data> as the root element, as shown below.

As can be seen, the <dict_data> has two main child elements, one for storing the dictionary data's default attributes (<dict_default_attribute>), and one to store the data itself (<dict_body>).

10.6.3.2 Dictionary default attributes <dict_default_attribute>

Dictionary default attributes store parameters to use when displaying the dictionary data. Be aware that these parameters may not always be respected, if the viewer is not able to handle the values/behaviors set in this module, or if the viewer is configured to give precedence to user settings. In addition, as was explained in 10.3.2.3, some of the parameters defined here may also be set in the whole text flow (in the <flow_default_attribute> child element of <flow_entry> see 10.3.2.4). These are used as default if the local ones (the ones we define here) are not set. In case both are defined, the local settings take precedence.

The <dict_default_attribute> tag has the following attributes and child elements.

[Attributes]

- baseline:** Defines the orientation of the baseline (and therefore of the text) for each flow. The possible values are listed below. May be omitted. If neither this attribute nor the corresponding option in the `<flow_default_attribute>` (see 10.3.2.3 for details) is set, the default value depends on the viewer.
- "right": The writing direction is horizontal (left to right). However, the direction can be changed at the user's option.
 - "right_only": The writing direction is horizontal (left to right), and the user cannot change the setting. However, if it is not supported by the viewer, this setting is not necessarily applied.
 - "down": The writing direction is vertical (top to bottom). However, the direction can be changed at the user's option.
 - "down_only": The writing direction is vertical (top to bottom), and the user cannot change the setting. However, if it is not supported by the viewer, this setting is not necessarily applied.
- valign:** Determines how the text box is to be positioned. The vertical position for horizontally written text, or the horizontal position for vertically written text is what is determined here. The value described below is eligible. When omitted, the text box should be displayed from the top of the display area.
- "middle": The content of the `<text_body>` tag is centered within the display area. See Figure 5 in 10.6.2.2. If the content is larger than the display area, this attribute is ignored,

[Child elements]

`<dict_default_font>` Defines the default font and font color for this dictionary data object. When omitted, defaults to the `<flow_default_attribute>` value. It has the following attribute.

[Attributes]

fontname: Default font name. More than one font may be specified. In that case, each font name should be separated by a comma (0x2c in Unicode). For instance:

fontname="Aaa sans serif,Bbb gothic"

The viewer should use the first listed font that is available. When omitted, defaults to the `<flow_default_attribute>` value.

color_space, color, opacity:

Defines the font color to be used for the content data. Written in the standard *color* data format. When omitted, defaults to the `<flow_default_attribute>` value.

ruby_flag: Defines whether ruby in the content data is to be viewed or not. When omitted, defaults to the `<flow_default_attribute>` value. The text to be displayed when ruby is turned on is the one included in the `<ruby>` tag of dictionary data object instances. The following values may be used.

"yes": Ruby should be displayed, but can be turned off by the user.

"yes_only": Ruby shall be displayed, and cannot be turned off by the user. However, this does not apply to viewers not able to display ruby.

“no”: It is recommended that ruby not be displayed, but can still be turned on according to user preferences.

“no_only”: Ruby shall not be displayed, and cannot be turned off by the user. However, this does not apply if the viewer is incapable of disabling ruby display.

<dict_default_background> Defines the background color and background image to be used for this dictionary data object. When both are defined, the image is drawn centered in the screen filled with the background color. Finally, the content of the **<dict_body>** tag is rendered on the top. If the background image is too large to fit inside the screen, it is scaled down with the aspect ratio being preserved. The behavior in case of omission of this tag should conform to the behavior specified in case of omission of its attributes. The following attributes and child elements are available.

[Attributes]

color_space, color, opacity:

Defines the color to be used as background color. Written in the standard *color* data format. When omitted, defaults to the **<flow_default_attribute>** value.

type: Stores the MIME type of the background image. May be omitted only if the **src** attribute is also omitted. The possible values are listed below.

"image/png"

"image/jpeg"

src: Sets the filename of the image to use as a background image, written in the standard *Filename* format. May be omitted. Before opening the file specified in this attribute, it shall be checked against the type attribute.

[Child elements]

<permission_info>:

Defines the permissions about the image referred to by the **src** attribute. Written in the standard *Permission* format. May be omitted. If the **src** attribute is omitted, this permission information should be ignored.

<dict_default_background_music>

Defines the background music to be played while displaying this dictionary data object. If this tag is omitted, nothing should be played. The following attributes and child elements are available.

[Attributes]

type: Stores the MIME type of the background music. Shall not be omitted. The possible values are listed below.

"audio/mp3"

src: Sets the filename of the background music, written in the standard *Filename* format. Shall not be omitted. Before opening the file specified in this attribute, it shall be checked against the type attribute.

loop: Specifies whether the background music should be played iteratively (i.e. repeated from the beginning every time the end is reached). Possible values are “yes” and “no”. In case of omission, no is assumed.

[Child elements]

<permission_info>:

Defines the permissions about the sound file referred to by the src attribute. Written in the standard *Permission* format. May be omitted.

<line_breaking_method> Specifies the algorithm to be used to determine how text should be split in lines. Various languages handle this in various ways, so this element allows for some flexibility. When omitted, defaults to the <flow_default_attribute> value. This tag has the following attributes and child elements.

[Attributes]

method: Chooses the line breaking method. Defaults to "none". Possible values are:

"none": no special processing, when a line is filled with characters, go to the next one.

Other values are possible for localization. See Clause A.2.

Other attributes may be added by localized methods. See Clause A.2.

[Child elements]

None, unless added by localized methods. See Clause A.2.

10.6.3.3 Main part of the dictionary data <dict_body>

Stores the main part of the dictionary data. It has the following child element.

[Child elements]

<dic-item> Records information for each individual entry. The same number of the tags exist as words defined. At least one instance of this tag shall appear in a content.

10.6.3.4 Entry item <dic-item>

<dic-item> has the following attributes and child elements.

[Attributes]

type: Specifies the type of the entry in *Standard character string*. May be omitted.

id: Specifies an identifier (Reference ID) for this entry. Reference ID is used for:

- a) accessing the item from the other parts of the content;
- b) identifying the item in editing and updating the content.

Shall not be omitted. It is an alphanumeric character string and may be an empty string ("").

rank Represents the importance level for this entry. (e.g. for learners) in numerals [0-9].

level Records the hierarchy of entries (e.g. chapter-section hierarchy), which is otherwise lost. To be specified in numerals [0-9]. Recorded in the order of importance (i.e. the smaller, the nearer to the top of the hierarchy) though it is not required that the values be consecutively used.)

page_break Inserts a page break. When omitted, it defaults to "no."
"yes": insert a page break

"no":	not insert a page break
turning_page_control	Defines how ordinary scrolling interacts with the page breaks, by allowing or forbidding crossing the page break forward or backward. The possibilities are listed below.
"on":	The page break shall not be crossed by scrolling.
"off":	The page break can be crossed by scrolling.
"forward":	The page break can be crossed by scrolling only when moving forward.
"back":	The page break can be crossed by scrolling only when moving backward.
When omitted, the behavior should conform to the global setting defined for the <i>dictionary data object instance</i> (the turning_page_control attribute of <flow_data>, see 10.3.2.4).	
revision	Specifies the revision number corresponding to the instance of this tag in a numeral not smaller than 1. May be omitted. When omitted, it is interpreted as no revision, i.e. the first version of the content.
delete	Specifies if this tag should be removed due to revision. The following values are permitted. Once a certain instance of this tag was deleted setting its delete attribute to "yes", a new instance of <dic-item> tag should be added when it is restored in later revisions. When omitted, it defaults to "no."
"yes":	deleted
"no":	not deleted

[Child elements]

<dic-item> can have the following tags that can appear in <text_body>

<p>, <scrolling_text>,
, <hr/>, , <i>, <u>, , <u>, <oblq>, <sc>, <mlg>, <horizontal>, <ruby>, <sub>, <sup>, <external_char>, <mask>, <char_id>, <head>, <object>, <div>, <table>...

Note the following:

- <head> shall appear exactly once at the top of inside <dic-item>;
- in <head>, <headword> shall appear at least once;
- if the <headword> tag appears more than once, bookmarks and word book functions are applicable to the first one only;
- <external_char> shall not appear inside <headword>.

[Child elements]

<dic-item> has the following child elements other than above.

<gender>	Gramatical gender info. <psp> can include such information without using this tag, depending the policy of the publisher.
<glab>	Denotes grammatical label (e.g. ":AmE", "Abbrev.").
<pronunciation>	Records a phonetic alphabet string. It has the following attributes. This tag is intended for recording phonetic alphabet strings outside <headword> and <subheadword> while such usage is not prohibited. The following attributes are possible.

[Attributes]

phonetic_notation: Denotes the phonetic alphabet used. Defaults to "IPA". The following values are pre-defined while other user-defined values are allowed as well. However, it is strongly recommended that "IPA" be used when reasonably applicable.

- <psp> Denotes a part of speech. Other grammatical marks indicating gender and case may be included here, depending on the policy of the publisher.
- <inflec> Denotes inflection of the word form.
- <slabel> Denotes category of the terms (e.g. "philosophy" , "commerce" and their abbreviations).
- <guideword> Also denotes the category of the terms. However, they are different from <slabel> in that the latter is mandatory when the (sub)headword falls into one of the applicable categories while the former is optional and intended to be used when they are necessary to distinguish multiple definitions.
- <spellout> Specifies the repeated string that is to be either replaced with other strings (possibly a symbol) that denotes omission or simply repeated.

[Attributes]

org specifies the string denoting omission. This tag specifies how omission should be represented in the final output.

Example:

```
<example><spellout org = "~">spell</spellout> one's name full
</example>
```



~ one's name full

- <variant> Denotes variants in a *Standard character string*.
- <etymology> Denotes etymological information in a *Standard character string*.

Example:

```
<dict_data>
<dict_default_attribute>...</dict_default_attribute>
<dict_body>
<dic-item page_break=" yes" turning_page_control="off">
<head>
<headword table_id="ST0001 ">やま【山】 </headword><key>やま</key>
<headword table_id="ST0001 ">やま【山】 </headword><key>山</key>
</head>
...
<dic-item>
<dic-item ...revision=" 2">...<dic-item>
<dic-item ...revision=" 1" delete=" yes">...<dic-item>
...
</dict_body>
</dict_data>
```

10.6.4 Image object instance

Regular image file referred to by the <object> tag of the *text object instance* (see 10.6.2.3). In the present state of this standard, it can either be a PNG file or a JPEG file.

10.6.5 Sound object instance

Regular sound file referred to by the <sound_object_entry> tag, or in the <flip_animation> tag. In the present state of this standard, MP3, AAC and SMAF files may be used.

10.6.6 Animation object instance

The following tags are eligible for animation object instances.

The image sequence animation <flip_animation> defines an animation as a sequence of images changing at fixed time intervals. Written as an XML file, as follows

Example:

```
<?xml version="1.0" encoding="Shift_JIS" ?>

<flip_animation renewal_time="500ms">
...
</flip_animation>
```

The <flip_animation> tag has the following attribute and child elements.

[Attribute]

renewal_time: Defines the interval between image switches, written in the standard *Time* format for durations up to 60 s. Defaults to "1s" when omitted. However, some viewers may not be able to respect the defined delay due to capability limitations.

[Child elements]

<flip_animation_sound> Defines a sound object that is to be played simultaneously to the animation. May be omitted. It has the following attributes.

[Attributes]

type: Records the MIME type of the sound file. In this standard, only "audio/mp3" is permitted. Shall not be omitted.

src: Name of the sound file, written in the standard *Filename* format. Shall not be omitted. Before opening the file specified in this attribute, it shall be checked against the type attribute.

<flip_animation_source> Each still image that composes the animation is recorded in separate instances of this tag. There shall be at least one instance of this tag. When there is more than one of this type of tag in the definition of an animation object, all the images referred to by the src attribute shall have the same size. It has the following attributes.

[Attributes]

type: Defines the type of image file registered by the src attribute as a MIME type. Shall not be omitted. In the present specification, the following formats are accepted:
"image/png"
"image/jpeg"

src: Name of the image file, written in the standard *Filename* format. Shall not be omitted. Before opening the file specified in this attribute, it shall be checked against the type attribute.

renewal_time: Time to wait before switching to the next image, written in the standard *Time* format, with a maximum of 60 s. If omitted, it defaults to the value set in the renewal_time attribute of <flip_animation>.

When the animation object is displayed in a loop (with the loop attribute of <object> set to more than "1"), the loop-time is determined as follows:

If the animation includes a sound file: The animation is considered over when all images have been shown and the associated sound has been played entirely. Repetition starts at this point. Note that the image sequence shall be shown again from the first image.

If the animation does not include a sound file: The animation simply starts again after it reached the end.

Example:

```
<flip_animation renewal_time="500ms" >
  <flip_animation_source src="aaa1.jpg" type="image/jpeg"/>
  <flip_animation_source src="aaa2.jpg" type="image/jpeg"/>
  ...
  <flip_animation_source src="aaan.jpg" type="image/jpeg" renewal_time="1s" />
</flip_animation>
```

10.6.7 Search page object instance <search_page>

A *search page object instance* is written as shown in the following example.

Example:

```
<?xml version="1.0" encoding="UTF-8" ?>

<search_page>
  <search_page_title type="image/png" src="image1.png">
    Foobar dictionary
  </search_page_title>
  <key_input_region table_id="ST0001" search_type="matches_first">
    ...
  </key_input_region>
  <key_input_region table_id="ST0002" search_type="match_only">
    ...
  </key_input_region>
  <search_link_item char_id="CR0001"> ... </search_link_item>
  <search_link_item char_id="CR0002"> ... </search_link_item>
  ...
</search_page>
```

A sample rendering is also shown in Figure 10. As can be seen, it is divided in 4 regions, as described below. How these regions are visually organized is dependant on the viewer. Some viewers may even decide not to display all of them.

- Title string: Region displaying the title of the search page. Defined in the `<search_page_title>` tag.
- Title image: Region displaying the search page's title image. Defined in the `<search_page_title>` tag.
- Keyword input field: Region to let the user input the keywords on which the search should be based. It is composed of actual input fields, as well as explanatory strings such as "Reading?" or "Spelling?". Each line is associated with a search table, against which the keywords are matched in order to produce the search result. Defined in the `<key_input_region>` tag.
- Link list: Displays a list of links to other search pages or specific parts of the documents. Defined in the `<search_link_item>` tag.

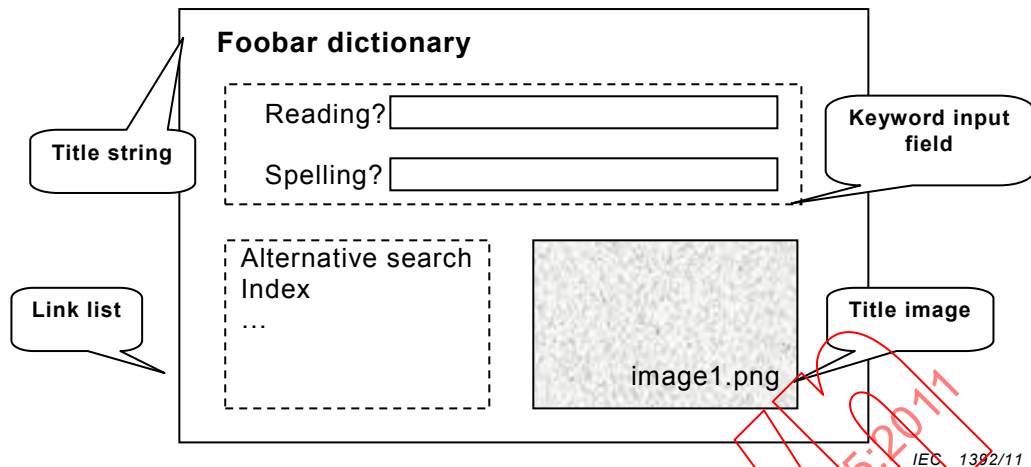


Figure 10 – Example of search page object instance rendering

The <search_page> tag has the following child elements.

[Child elements]

<search_page_title> Defines the title of the search screen. May be omitted. It has the following attributes and child elements.

[Attributes]

type: Stores the MIME type of the image to use as a title image. May be omitted only if the src attribute is also omitted. The possible values are listed below.

"image/png"

"image/jpeg"

src: Sets the filename of the image to use as a title image, written in the standard *Filename* format. May be omitted. Before opening the file specified in this attribute, it shall be checked against the type attribute.

[Child elements]

The string to be used as a title is written inside the tag as an *External extended character string*.

Example:

```
<search_page_title type="image/jpeg" src="image.jpg">
  Foobar dictionary
</search_page_title>
```

<key_input_region> Defines the keyword input region. Within a single search screen, there can only be one or two instances of this element. The viewer uses this information to display the input region, and then conducts the search based on user input. It has the following attributes and child elements.

[Attributes]

- table_id:** Records the ID number of the search table to be used as a base for the search. It corresponds to the number defined in <search_table_def> (see 10.3.2.6). Not to be confused with Reference ID (see 10.6.3.4). Shall not be omitted
- search_type:** Defines the type of search, to be chosen from the possibilities listed below. May be omitted, and defaults to "matches_only" in that case.

"matches_only": "Matches only" search mode. The normalized (see 10.3.2.6, <key_normalization>) keyword is forward-matched to each entry word in the search table designated by the table_id attribute. The result of this search is all the words that match using those criteria, and only those words.

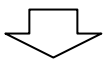
Sample results for "dict"



- dictate
- dictation
- dictator
- dictatorial
- dictatorship
- diction
- dictionary
- dictum

"matches_first": "Matches_first" mode. The comparison method is the same as in "maches_only", but the result list is a little different. It contains all the words matching the normalized keyword, and also the following entries in the search table.

Sample results for "dict"



- dictate
- dictation
- dictator
- dictatorial
- dictatorship
- diction
- dictionary
- dictum
- didactic
- didactical
- :

[Child elements]

<key_input_region_prompt> Records the character string used as a label to explain the keyword input field, such as the "Reading?" or "Spelling?" shown in Figure 10. Shall not be omitted.

[Child elements]

The character string to be used as explained above.

<enable_key_type> Defines what kind of character is allowed in the keyword input field. The attributes and child elements of this tag are the same as those used in search table data's <enable_key_type> (see 10.3.2.6), therefore the details are not repeated here. However, the character types allowed in the search page object's <enable_key_type> shall be a subset of those defined in the associated search table. Shall not be omitted.

Example:

```
<key_input_region table_id="ST0001" search_type="matches_first">
  <key_input_region_prompt>Reading?</key_input_region_prompt>
  <enable_key_type number="no" alphabet="yes" kana="yes"/>
</key_input_region>

<key_input_region table_id="ST0002" search_type="match_only">
  <key_input_region_prompt>Spelling?</key_input_region_prompt>
  <enable_key_type number="no" alphabet="no" kana="yes"/>
</key_input_region>
```

<search_link_item> Records the information defining the link list. They may point to other search tables, or to a specific position in a text flow. The jumps are handled by using the <trigger_pointer> (see 10.4.3) and <action_page_jump> (see 10.4.4.3) tags. They should be recorded in the event data (see 10.4) of the flow hosting this search page object. Shall not be omitted. It has the following attribute and child element.

[Attribute]

char_id: Defines the ID number of this search link item, written in the standard *char ID* format. Shall not be omitted.

[Child elements]

<search_link_title> Sets the title of this link, that is, the text that will be displayed and act as the link. It is written as an *External extended character string*. Shall not be omitted.

Example:

```
<search_link_item char_id="CR0001">
  <search_link_title>Alternate search</search_link_title>
</search_link_item>

<search_link_item char_id="CR0002">
  <search_link_title>Index</search_link_title>
</search_link_item>
```

10.6.8 Movie object instance

Regular movie file referred to by the <movie_object_entry> tag (see 10.5.5). In the present state of this standard, only 3GPP2 files are allowed.

Withdrawing
IECNORM.COM: Click to view the full PDF of IEC 62605:2011

Annex A (normative)

Color names

A.1 Available color names

The color names that may be used in the attributes specifying color types are listed in Table A.1, with the corresponding RGB values.

Table A.1 – Color names

Color name	White	Black	Red	Green	Blue
RGB Value	#FFFFFF	#000000	#FF0000	#008000	#0000FF
Color name	Yellow	Purple	Aqua	Maroon	Navy
RGB Value	#FFFF00	#800080	#00FFFF	#800000	#000080
Color name	Olive	Teal	Gray	Silver	SlateBlue
RGB Value	#808000	#008080	#808080	#C0C0C0	#6A5ACD
Color name	MediumBlue	RoyalBlue	DodgerBlue	SkyBlue	SteelBlue
RGB Value	#0000CD	#4169E1	#1E90FF	#87CEEB	#4682B4
Color name	LightBlue	PaleTurquoise	Turquoise	Cyan	LightCyan
RGB Value	#ADD8E6	#AFEEEE	#40E0D0	#00FFFF	#E0FFFF
Color name	Aquamarine	DarkGreen	SeaGreen	LightGreen	Chartreuse
RGB Value	#7FFFD4	#006400	#2E8B57	#90EE90	#7FFF00
Color name	GreenYellow	LimeGreen	YellowGreen	OliveDrab	DarkKhaki
RGB Value	#ADFF2F	#32CD32	#9ACD32	#6B8E23	#BDB76B
Color name	PaleGoldenrod	LightYellow	Gold	Goldenrod	DarkGoldenrod
RGB Value	#EE8AA	#FFFFE0	#FFD700	#DAA520	#B8860B
Color name	RosyBrown	IndianRed	SaddleBrown	Sienna	Peru
RGB Value	#BC8F8F	#CD5C5C	#8B4513	#A0522D	#CD853F
Color name	Burlywood	Beige	Wheat	SandyBrown	Tan
RGB Value	#DEB887	#F5F5DC	#F5DEB3	#F4A460	#D2B48C
Color name	Chocolate	Firebrick	Brown	Salmon	Orange
RGB Value	#D2691E	#B22222	#A52A2A	#FA8072	#FFA500
Color name	Coral	Tomato	HotPink	Pink	DeepPink
RGB Value	#FF7F50	#FF6347	#FF69B4	#FFC0CB	#FF1493
Color name	PaleVioletRed	Magenta	Violet	Plum	Orchid
RGB Value	#DB7093	#FF00FF	#EE82EE	#DDA0DD	#DA70D6
Color name	DarkViolet	BlueViolet	MediumPurple	Thistle	Lavender
RGB Value	#9400D3	#8A2BE2	#9370DB	#D8BFD8	#E6E6FA
Color name	MistyRose	Ivory	LemonChiffon	Moccasin	
RGB Value	#FFE4E1	#FFFFFF	#FFFACD	#FFE4B5	

A.2 Localization

A.2.1 Possible additions

The proposals made here for localization are not normative. They are given as examples of additions that should be made to the format before use for a particular language. This localization part is subject to standardization on a language-specific basis when the specification is put to use for that particular language.

A.2.2 Standard characters

For localization purposes, the character sets listed in Table A.2 may be used for the standard character set of the document, depending on the language.

Table A.2 – Examples of additional standard character sets

Target language	Name	Description
Japanese	"JIS X 0201, JIS X 0208:1997"	Characters in the range of Shift JIS
English	"ISO 646-IRV"	The 7bit ASCII characters
French	"ISO 8859-15"	Characters in the range of ISO 8859-15 (Latin-9)

A.2.3 Characters usable for reading

In addition to the basic characters allowed in the *Reading* data format, the language-specific extensions listed in Table A.3 are proposed.

Table A.3 – Example of additional characters usable for readings

Target language	Name	Description
Japanese	Full-width Japanese katakana characters	アアイウエオカガキクグケゴコサザシスズセソゾタチヂッツツテドト ドナニヌネノハバヒビフブヘベホボマミムメモヤユヨヨラリルレロワウキ エランヴカケ (0x30A1 to 0x30F6)
Japanese	Japanese long vowel mark	ー (0x30FC)
French	French alphabet extensions	Â (0x00C0) Ã (0x00C2) Æ (0x00C6) Ç (0x00C7) È (0x00C8) É (0x00C9) Ê (0x00CA) Ë (0x00CB) Î (0x00CE) Ï (0x00CF) Ô (0x00D4) Ö (0x00D6) Ù (0x00D9) Ú (0x00DB) Ü (0x00DC) Ý (0x00DF) à (0x00E0) â (0x00E2) æ (0x00E6) ç (0x00E7) è (0x00E8) é (0x00E9) ê (0x00EA) ë (0x00EB) î (0x00EE) ï (0x00EF) ô (0x00F4) ù (0x00F9) û (0x00FB) ü (0x00FC) ÿ (0x00FF) Œ (0x0152) œ (0x0153)

A.2.4 Line breaking methods

A.2.4.1 For Japanese

Possible value for the method attribute of the `<flow_default_line_breaking_method>` (see 10.3.2.3) and `<line_breaking_method>` (see 10.6.2.2) tags.

"run_down": move characters from the end of a line to the beginning of the next one, to respect the position restrictions expressed by the `<top_prohibit_char>` and `<end_prohibit_char>` tags.

Additional attributes of the `<flow_default_line_breaking_method>` and `<line_breaking_method>` tags when using the "run_down" method.

hanging_punctuation: Activates or deactivates the processing of hanging characters. Possible values are "yes" and "no". Defaults to "yes" when method is set to "run_down". When activated, the characters listed in `<hanging_char>`, instead of being displayed at the beginning of a line, should be displayed after the end (i.e. in the right margin) of the previous line. Does not apply if the character is the first of a paragraph.

Additional child elements of the `<flow_default_line_breaking_method>` and `<line_breaking_method>` tags when using the "run_down" method.

`<top_prohibit_char>` Lists the characters that must not appear at the beginning of a line, except as the first character of the paragraph. Listed as an *Extended character string*. Defaults to an empty list when omitted.

Example:

```
<top_prohibit_char>}}}</top_prohibit_char>
```

`<end_prohibit_char>` Lists the characters that must not appear at the end of a line, except as the last letter of the paragraph. Listed as an *Extended character string*. Defaults to an empty list when omitted.

Example:

```
<end_prohibit_char>({{"</end_prohibit_char>
```

`<hanging_char>` Lists the characters to be displayed as hanging punctuation. Listed as an *Extended character string*. Defaults to an empty list when omitted.

Example:

```
<hanging_char>、 。 ! ? : ; <hanging_char>
```

A.2.4.2 For English

Possible value for the method attribute of the `<flow_default_line_breaking_method>` and `<line_breaking_method>` tags.

"word_wrap": Line breaks can only occur on space characters, unless that would cause text to overflow the line.

A.2.5 Sorting rules for <search_table_def>

The methods listed in Table A.4 are proposed.

Table A.4 – Example of additional sorting rules

Target language	Sorting_rule name	Description
Japanese	shift_jis	Characters are ranked according to their code points in the Shift_JIS encoding. Rank for characters not covered by this encoding is unspecified.
English	en	Characters are ranked according to their code points in ASCII encoding. Rank for characters not covered by this encoding is unspecified.
French	fr	Characters are sorted in the following order: A Â Ã Ä Å Æ B C Ç È É Ê Ë Ì Í Î Ï J K L M N O Ö Ø Æ P Q R S T U Û Ü V W X Y Z The same order is valid for lower case letters too. Rank for other characters is unspecified, and may simply follow the encoding in use, or the Unicode code points.

A.2.6 Additional attributes for <enable_key_type>

It is proposed that <enable_key_type> have the language-specific attributes listed in Table A.5 in addition to those described in 10.3.2.6. Each accepts 2 values; "yes" and "no". Defaults to "no" when omitted.

Table A.5 – Example of additional language specific attributes for <enable_key_type>

Target language	Character type name	Corresponding characters (all values are in Unicode)			
Japanese	fullwidth_numerals	0 to 9 (0xFF10 to 0xFF19)			
Japanese	fullwidth_alphabet	A to Z (0xFF21 to 0xFF3A) and a to z (0xFF41 to 0xFF5A)			
Japanese	kana	Half-width katakana: アイウエオカキクケコサシスセソタチツテトナニヌネノハヒフヘホマミメモヤユヨラリルレロワヰ (0xFF71 to 0xFF9F) and アィウヱォカギクグケゲコゴサザシスズセゼソダヂチツツテテトド ナニヌネノハババヒビビフブフヘベホボボマミムメモヤヤユユヨヨラリルレロワワヰヰ ヲ (0x30A1 to 0x30F6) Full-width katakana: アァイィウゥエェオォカガキギクグケゲコゴサザシスズセゼソダヂチツツテテトド ナニヌネノハババヒビビフブフヘベホボボマミムメモヤヤユユヨヨラリルレロワワヰヰ ヲ (0x3041 to 0x3093) Full-width hiragana: ああいうえおおががきぎくぐけげこごさししじすずせぜそぞただちぢっつづてとど なにぬねのはばびびびぶぶへべほほまみむめもやゆよよりるれろわわゐゑを ん (0x3041 to 0x3093) Long vowels: ー (0x30FC, full-width) and - (0xFF70, half-width)			
	kanji	To be appropriately defined e.g. kanji characters included in JIS X 0208:1997 or JIS X 0213:2004			
French	French_alphabet_extensions	A (0x00C0)	À (0x00C2)	Æ (0x00C6)	Ç (0x00C7)
		É (0x00C8)	Ê (0x00CA)	Ë (0x00CB)	
		Î (0x00CE)	Ï (0x00CF)	Ô (0x00D4)	Ö (0x00D6)
		Û (0x00D9)	Ü (0x00DB)	Û (0x00DC)	Ÿ (0x00DF)
		à (0x00E0)	â (0x00E2)	æ (0x00E6)	ç (0x00E7)
		è (0x00E8)	é (0x00E9)	ê (0x00EA)	ë (0x00EB)
		î (0x00EE)	ï (0x00EF)	ô (0x00F4)	ù (0x00F9)
		û (0x00FB)	ü (0x00FC)	ÿ (0x00FF)	œ (0x0152)
		œ (0x0153)			

A.2.7 Normalization methods for <key_normalization>

A.2.7.1 For Japanese language

Additional attributes of the <key_normalization> tag (see 10.3.2.6).

- cho_on: Japanese long vowel (cho on) conversion method. Possible values are "delete" (remove the character), "repeat" (repeat the vowel preceding the character) and "no" (do nothing). Defaults to "delete".
- daku_on: Japanese voiced sound (daku on) conversion. Possible values are "yes" (convert to unvoiced, or sei on) and "no" (no conversion). Defaults to "no".

- handaku_on:** Japanese semi-voiced sound (handaku on) conversion. Possible values are "yes" (convert to unvoiced) and "no" (no conversion). Defaults to "no".
- soku_on:** Japanese geminate consonant (soku on or "small tsu") conversion. Possible values are "yes" (convert to large character) and "no" (no conversion). Defaults to "no".
- yo_on:** Japanese palatalization (yo on) conversion. Possible values are "yes" (conversion to large character) or "no" (no conversion). Defaults to "no".
- other_small_kana:** Other Japanese small character (hiragana and katakana) conversion. Possible values are "yes" (convert to large character) and "no" (do not convert). Defaults to "no".

In addition to the methods controlled by the preceding parameters, the following normalization process should be applied all the time.

All alphanumeric characters, both full-width and half-width, should be normalized to half-width. All Japanese hiragana and katakana, both full-width and half-width, should be normalized to full-width katakana.

A.2.7.2 For French language

Additional attributes of the <key_normalization> tag.

- diacritic_removal:** drops all diacritics (accents, tildas, and cedilla) from the letter that carry them. Possible values are "yes" and "no". Defaults to "no".

A.2.8 Character encoding conversions

Converting a piece of text from one character encoding to another can bring some ambiguities, since different encodings, even for the same language, will sometimes cover slightly different character sets. During the localization process, such ambiguities should be considered, and the preferred conversion schemes should be specified.

The W3C's XML Japanese profile (<http://www.w3.org/TR/japanese-xml/>) is a good example of referencing such ambiguities in the Japanese language.

A.3 Adaptation

This standard specifies only the most basic media files as usable. It is to be understood that other media files may be used with media types and other values appropriately defined as needed.

A.4 Specification of the XMDF-LeXML format in the RELAX NG syntax

The syntax of the XMDF-LeXML format is formally given here as a compact RELAX NG schema. Note that it does not include any localization-related addition.

namespace a = "http://relaxng.org/ns/compatibility/annotations/1.0"

Yes_No = "yes" | "no"

Trigger_List = trigger_pointer

Action_List = action_play | action_page_jump

TextWithGaiji = (text | external_char)*

FiveSize = "maximum" | "big" | "medium" | "small" | "minimum"

BaseLine = "right" | "right_only" | "down" | "down_only"

ViewType = "portrait" | "portrait_only" | "landscape" | "landscape_only"

HoldFlag = "scope" | "on_power" | "save"

Turn_Page_Val = "on" | "off" | "forward" | "back"

personal_name =

element personal_name {

attlist_personal_name,

((first_name?, middle_name?, last_name?)

| (first_name?, last_name?, middle_name?)

| (last_name?, first_name?, middle_name?)

| (last_name?, middle_name?, first_name?))

}

attlist_personal_name &= empty

first_name = element first_name { attlist_first_name, TextWithGaiji }

attlist_first_name &= attribute reading { text }?

middle_name = element middle_name { attlist_middle_name, TextWithGaiji }

attlist_middle_name &= attribute reading { text }?

last_name = element last_name { attlist_last_name, TextWithGaiji }

attlist_last_name &= attribute reading { text }?

organization_name =

element organization_name { attlist_organization_name, TextWithGaiji }

attlist_organization_name &= attribute reading { text }?

address_info =

```
element address_info {  
    attlist_address_info,  
    postal_code?,  
    address?,  
    telephone?,  
    fax?,  
    mail_address?,  
    website?,  
    address_other_info?  
}  
  
attlist_address_info &= empty  
postal_code = element postal_code { attlist_postal_code, text }  
attlist_postal_code &= empty  
address = element address { attlist_address, TextWithGaiji }  
attlist_address &= empty  
telephone = element telephone { attlist_telephone, text }  
attlist_telephone &= empty  
fax = element fax { attlist_fax, text }  
attlist_fax &= empty  
mail_address = element mail_address { attlist_mail_address, text }  
attlist_mail_address &= empty  
website = element website { attlist_website, text }  
attlist_website &= empty  
address_other_info =  
    element address_other_info {  
        attlist_address_other_info, TextWithGaiji  
    }  
}
```

attlist_address_other_info &=

[a:defaultValue = "preserve"]

attribute xml:space { "default" | "preserve" }?

permission_info =

element permission_info {

attlist_permission_info, print_permission?, copy_permission?

}

attlist_permission_info &= empty

print_permission =

element print_permission { attlist_print_permission, empty }

attlist_print_permission &=

[a:defaultValue = "no"]

attribute permission { "authorized" | "no" }?

copy_permission =

element copy_permission { attlist_copy_permission, empty }

attlist_copy_permission &=

[a:defaultValue = "no"]

attribute permission { "authorized" | "in_device_only" | "no" }?

keyword_list = element keyword_list { attlist_keyword_list, keyword+ }

attlist_keyword_list &= empty

keyword = element keyword { attlist_keyword, TextWithGaiji }

attlist_keyword &=

attribute category { text }?,

attribute reading { text }?

vertex = element vertex { attlist_vertex, empty }

attlist_vertex &= attribute position { text }

bvf = element bvf { attlist_bvf, book_info, body_module, parts_module }

attlist_bvf &=

attribute id_type { text }?,

attribute id { text }?,

attribute default_ccs { text },

attribute display_size { text }?

book_info =

element book_info {

attlist_book_info,

title_info,

author_info?,

publisher_info?,

seller_info?,

book_id_info?,

classification_info?,

rating?,

publication_place?,

publication_date_info?,

net_price_info?,

book_abstract?,

front_cover_image?,

spine_cover_image?,

keyword_list?,

```

    other_book_info?

}

attlist_book_info &= empty

title_info =

    element title_info {

        attlist_title_info,

        series_title?,

        title,

        subtitle?,

        edition_info?,

        title_other_info?

    }

attlist_title_info &= empty

series_title =

    element series_title { attlist_series_title, TextWithGaiji }

attlist_series_title &= attribute reading { text }?

title = element title { attlist_title, TextWithGaiji }

attlist_title &= attribute reading { text }?

subtitle = element subtitle { attlist_subtitle, TextWithGaiji }

attlist_subtitle &= attribute reading { text }?

edition_info =

    element edition_info { attlist_edition_info, TextWithGaiji }

attlist_edition_info &=

    attribute this_version { text }?,

    [ a:defaultValue = "preserve" ]

    attribute xml:space { "default" | "preserve" }?

```

title_other_info =

element title_other_info { attlist_title_other_info, TextWithGaiji }

attlist_title_other_info &=

[a:defaultValue = "preserve"]

attribute xml:space { "default" | "preserve" }?

author_info = element author_info { attlist_author_info, author+ }

attlist_author_info &= empty

author =

element author {

attlist_author,

(personal_name | organization_name),

address_info?,

author_other_info?

}

attlist_author &=

[a:defaultValue = "author"]

attribute role {

"author"

| "editor"

| "translator"

| "supervisor"

| "designer"

| "photographer"

| "illustrator"

| "binder"

| "planner"

| "other"

}?

author_other_info =

element author_other_info { attlist_author_other_info, TextWithGaiji }

attlist_author_other_info &=

[a:defaultValue = "preserve"]

attribute xml:space { "default" | "preserve" }?

publisher_info =

element publisher_info {

attlist_publisher_info,

((publisher, publisher_office) | publisher | publisher_office),

publisher_other_info?

}

attlist_publisher_info &= empty

publisher =

element publisher { attlist_publisher, publisher_name, address_info? }

attlist_publisher &= empty

publisher_name =

element publisher_name { attlist_publisher_name, TextWithGaiji }

attlist_publisher_name &= attribute reading { text }?

publisher_office =

element publisher_office {

attlist_publisher_office, organization_name, address_info?

}

attlist_publisher_office &= attribute publisher_code { text }?

publisher_other_info =

```
element publisher_other_info {  
    attlist_publisher_other_info, TextWithGaiji  
}
```

attlist_publisher_other_info &=

```
[ a:defaultValue = "preserve" ]  
attribute xml:space { "default" | "preserve" }?
```

seller_info =

```
element seller_info {  
    attlist_seller_info,  
    ((seller, seller_office) | seller | seller_office),  
    seller_other_info?  
}
```

attlist_seller_info &= empty

seller = element seller { attlist_seller, seller_name, address_info? }

attlist_seller &= empty

seller_name = element seller_name { attlist_seller_name, TextWithGaiji }

attlist_seller_name &= attribute reading { text }?

seller_office =

```
element seller_office {  
    attlist_seller_office, organization_name, address_info?  
}
```

attlist_seller_office &= attribute seller_code { text }?

seller_other_info =

element seller_other_info { attlist_seller_other_info, TextWithGaiji }

attlist_seller_other_info &=

[a:defaultValue = "preserve"]

attribute xml:space { "default" | "preserve" }?

book_id_info = element book_id_info { attlist_book_id_info, book_id+ }

attlist_book_id_info &= empty

book_id = element book_id { attlist_book_id, text }

attlist_book_id &= attribute type { text }

classification_info =

element classification_info {

attlist_classification_info, classification+

}

attlist_classification_info &= empty

classification =

element classification { attlist_classification, TextWithGaiji }

attlist_classification &= attribute type { text }

rating = element rating { attlist_rating, empty }

attlist_rating &=

[a:defaultValue = "no"] attribute adult { Yes_No }?,

[a:defaultValue = "no"] attribute violence { Yes_No }?

publication_place =

element publication_place { attlist_publication_place, text }

attlist_publication_place &= empty

publication_date_info =

```

    element publication_date_info {
        attlist_publication_date_info, publication_date+
    }

```

attlist_publication_date_info &= empty

publication_date =

```

    element publication_date { attlist_publication_date, text }

```

attlist_publication_date &=

```

    [ a:defaultValue = "publish" ] attribute type { "publish" | "sale" }?

```

net_price_info =

```

    element net_price_info { attlist_net_price_info, net_price+ }

```

attlist_net_price_info &= empty

net_price = element net_price { attlist_net_price, text }

attlist_net_price &=

```

    attribute country { text }?,

```

```

    attribute unit { text },

```

```

    attribute other_info { text }?

```

book_abstract =

```

    element book_abstract { attlist_book_abstract, TextWithGaiji }

```

attlist_book_abstract &=

```

    [ a:defaultValue = "preserve" ]

```

```

    attribute xml:space { "default" | "preserve" }?

```

front_cover_image =

```

    element front_cover_image { attlist_front_cover_image, text }

```

attlist_front_cover_image &= attribute type { text }

spine_cover_image =

```

element spine_cover_image { attlist_spine_cover_image, text }
attlist_spine_cover_image &= attribute type { text }

```

```

other_book_info =

```

```

    element other_book_info { attlist_other_book_info, TextWithGaiji }
attlist_other_book_info &=
[ a:defaultValue = "preserve" ]
attribute xml:space { "default" | "preserve" }?

```

```

body_module =

```

```

    element body_module { attlist_body_module, flow_type_body }
attlist_body_module &= empty

```

```

flow_type_body =

```

```

    element flow_type_body {
        attlist_flow_type_body,
        flow_entry,
        special_page_link?,
        search_table?
    }

```

```

attlist_flow_type_body &= empty

```

```

flow_entry =

```

```

    element flow_entry {
        attlist_flow_entry, flow_default_attribute?, flow_data+
    }

```

```

attlist_flow_entry &= empty

```

flow_default_attribute =

```
element flow_default_attribute {  
    attlist_flow_default_attribute,  
    flow_default_size?,  
    flow_default_font?,  
    flow_default_background?,  
    flow_default_line_breaking_method?  
}
```

attlist_flow_default_attribute &=

```
attribute baseline { BaseLine }?,  
attribute view_type { ViewType }?,  
attribute lang { text }?,  
[ a:defaultValue = "ISO 639-3" ] attribute lang_code { text }?,  
[ a:defaultValue = "IPA" ] attribute phonetic_notation { text }?
```

flow_default_size =

```
element flow_default_size { attlist_flow_default_size, empty }
```

attlist_flow_default_size &=

```
attribute letter_spacing { FiveSize }?,  
attribute line_pitch { FiveSize }?,  
attribute margin { "big" | "medium" | "small" }?
```

flow_default_font =

```
element flow_default_font { attlist_flow_default_font, empty }
```

attlist_flow_default_font &=

```
attribute fontname { text }?,  
attribute fontsize { FiveSize }?,  
attribute bold_flag { Yes_No }?,
```

```

attribute color_space { "RGB" }?,
attribute opacity { "100" }?,
attribute color { text }?,
attribute ruby_flag { "yes" | "yes_only" | "no" | "no_only" }?,
attribute italic { Yes_No }?,
attribute oblique { Yes_No }?,
attribute small_caps { Yes_No }?,
attribute family {
    "monospace" | "sans-serif" | "serif" | "cursive" | "fantasy"
}?,
attribute ul_type { "rightscore" | "leftscore" | "throughscore" }?,
attribute em_type {
    "rightscore"
    | "leftscore"
    | "throughscore"
    | "kendot"
    | "bold"
    | "italic"
    | "bold_italic"
    | "reverse"
    | "shade"
}?,
flow_default_background =
    element flow_default_background {
        attlist_flow_default_background, empty
    }
attlist_flow_default_background &=

```

attribute color_space { "RGB" }?,

attribute opacity { "100" }?,

attribute color { text }?

flow_default_line_breaking_method =

element flow_default_line_breaking_method {

attlist_flow_default_line_breaking_method, empty

}

attlist_flow_default_line_breaking_method &=

[a:defaultValue = "none"] attribute method { "none" }?

flow_default_delimiter =

element flow_default_delimiter {

attlist_flow_default_delimiter, delimiter?

}

attlist_flow_default_delimiter &= empty

delimiter = element delimiter { attlist_delimiter, empty }

attlist_delimiter &=

attribute tag { text },

attribute type { text }?,

[a:defaultValue = ""] attribute open { text }?,

[a:defaultValue = ""] attribute close { text }?

flow_data = element flow_data { attlist_flow_data, event_info? }

attlist_flow_data &=

attribute flow_id { text }?,

attribute body_id { text },

[a:defaultValue = "off"]

attribute turning_page_control { Turn_Page_Val }?

special_page_link =

element special_page_link { attlist_special_page_link, special_page+ }

attlist_special_page_link &= empty

special_page = element special_page { attlist_special_page, text }

attlist_special_page &=

[a:defaultValue = "other"]

attribute kind {

"cover"

| "title_page"

| "preface"

| "contents"

| "body"

| "column"

| "note"

| "figure"

| "ad"

| "afterword"

| "appendix"

| "answer"

| "glossary"

| "bibliography"

| "commentary"

| "index"

| "imprint"

| "author_info"

| "other"

}?,

attribute title { text }?

search_table =

element search_table { attlist_search_table, search_table_def+ }

attlist_search_table &=

[a:defaultValue = "no"] attribute bookmark { Yes_No }?,

[a:defaultValue = "no"] attribute wordbook { Yes_No }?,

[a:defaultValue = "no"] attribute jump_search_root { Yes_No }?,

[a:defaultValue = "no"] attribute jump_search { Yes_No }?,

[a:defaultValue = "no"] attribute all_search { Yes_No }?

search_table_def =

element search_table_def {

attlist_search_table_def, enable_key_type, key_normalization

}

attlist_search_table_def &=

attribute id { text },

[a:defaultValue = "no"] attribute use_default { Yes_No }?,

[a:defaultValue = "unicode"] attribute sorting_rule { "unicode" }?,

attribute name { text },

attribute short_name { text },

attribute wild { Yes_No },

attribute blank { Yes_No },

attribute end { Yes_No },

attribute help_page_id { text }?

enable_key_type =

element enable_key_type { attlist_enable_key_type, empty }

attlist_enable_key_type &=

[a:defaultValue = "no"] attribute numerals { Yes_No }?,

[a:defaultValue = "no"] attribute basic_alphabet { Yes_No }?

key_normalization =

element key_normalization { attlist_key_normalization, empty }

attlist_key_normalization &=

[a:defaultValue = "yes"] attribute capitalization { Yes_No }?

event_info = element event_info { attlist_event_info, event+ }

attlist_event_info &=

[a:defaultValue = "single"] attribute display_type { "single" }?

event = element event { attlist_event, trigger, action }

attlist_event &= empty

trigger = element trigger { attlist_trigger, Trigger_List }

attlist_trigger &= empty

action = element action { attlist_action, Action_List }

attlist_action &=

[a:defaultValue = "sequential"]

attribute action_flag { "sequential" }?

trigger_pointer =

element trigger_pointer { attlist_trigger_pointer, pointer_region? }

attlist_trigger_pointer &=

attribute id { text },

[a:defaultValue = "click"] attribute action_flag { "click" }?

pointer_region =

element pointer_region { attlist_pointer_region, vertex+ }

attlist_pointer_region &= empty

action_play = element action_play { attlist_action_play, empty }

attlist_action_play &=

attribute object_id { text },

[a:defaultValue = "normal"] attribute action { "normal" }?

action_page_jump =

element action_page_jump { attlist_action_page_jump, empty }

attlist_action_page_jump &=

attribute book { text }?,

attribute book_type { text }?,

attribute page_id { text }?,

attribute center { text }?

parts_module =

element parts_module { attlist_parts_module, object_table }

attlist_parts_module &= empty

object_table =

element object_table {

attlist_object_table,

(dynamic_text_object_entry

| sound_object_entry

| search_page_object_entry

| movie_object_entry

| dict_data_object_entry)+

}

attlist_object_table &= empty

dynamic_text_object_entry =

```
element dynamic_text_object_entry {  
    attlist_dynamic_text_object_entry, permission_info?  
}
```

attlist_dynamic_text_object_entry &=

```
attribute src { text },  
attribute type { text },  
attribute object_id { text }
```

sound_object_entry =

```
element sound_object_entry {  
    attlist_sound_object_entry, sound_object_info  
}
```

attlist_sound_object_entry &=

```
attribute src { text },  
attribute type { text }
```

sound_object_info =

```
element sound_object_info { attlist_sound_object_info, empty }
```

attlist_sound_object_info &= attribute object_id { text }

search_page_object_entry =

```
element search_page_object_entry {  
    attlist_search_page_object_entry, permission_info?  
}
```

attlist_search_page_object_entry &=

```
attribute src { text },
```

attribute type { text },

attribute object_id { text }

movie_object_entry =

element movie_object_entry { attlist_movie_object_entry, empty }

attlist_movie_object_entry &=

attribute src { text },

attribute type { text },

attribute object_id { text }

dict_data_object_entry =

element dict_data_object_entry {

attlist_dict_data_object_entry, permission_info?

}

attlist_dict_data_object_entry &=

attribute src { text },

attribute type { text },

attribute object_id { text }

text_data =

element text_data {

attlist_text_data, text_default_attribute, text_body

}

attlist_text_data &= empty

text_default_attribute =

element text_default_attribute {

```

    attlist_text_default_attribute,

    text_default_font?,

    text_default_background?,

    text_default_background_music?,

    line_breaking_method?

}

attlist_text_default_attribute &=

    attribute baseline { BaseLine }?,

    attribute valign { "middle" }?

text_default_font =

    element text_default_font { attlist_text_default_font, empty }

attlist_text_default_font &=

    attribute fontname { text }?,

    attribute color_space { "RGB" }?,

    attribute opacity { "100" }?,

    attribute color { text }?,

    attribute ruby_flag { "yes" | "yes_only" | "no" | "no_only" }?

text_default_background =

    element text_default_background {

        attlist_text_default_background, permission_info?

    }

attlist_text_default_background &=

    attribute color_space { "RGB" }?,

    attribute opacity { "100" }?,

    attribute color { text }?,

    attribute type { text }?,

    attribute src { text }?

```

text_default_background_music =

element text_default_background_music {

attlist_text_default_background_music, permission_info?

}

attlist_text_default_background_music &=

attribute type { text },

attribute src { text },

[a:defaultValue = "no"] attribute loop { Yes_No }?

line_breaking_method =

element line_breaking_method { attlist_line_breaking_method, empty }

attlist_line_breaking_method &=

[a:defaultValue = "none"] attribute method { "none" }?

Inline =

text

| br

| hr

| font

| horizontal

| ruby

| external_char

| mask

| char_id

| i

| u

| em

| oblq

| sc

| mlg

| sub

| sup

| meaning

| example

| column

| \div

| span

| table

| select

| tts

| a

| object

All_tag_p = Inline | p | scrolling_text

All_tag_di = All_tag_p | head | subhead

All_tag = All_tag_di | page_break

text_body = element text_body { attlist_text_body, All_tag* }

attlist_text_body &= empty

p = element p { attlist_p, All_tag_p* }

attlist_p &=

attribute top_line_indent { text }?,

attribute top { text }?,

attribute bottom { text }?,

attribute align { "center" | "right" | "left" }?,

attribute drop_cap { text }?

scrolling_text =

element scrolling_text {

attlist_scrolling_text,

(text

| external_char

| font

| horizontal

| ruby

| sub

| sup

| object)*

}

attlist_scrolling_text &= empty

page_break = element page_break { attlist_page_break, empty }

attlist_page_break &= attribute turning_page_control { Turn_Page_Val }?

br = element br { attlist_br, empty }

attlist_br &= attribute clear { "left" | "right" | "all" }?

hr = element hr { attlist_hr, empty }

attlist_hr &=

attribute size { text }?,

attribute length { text }?,

attribute align { "left" | "center" | "right" }?

font = element font { attlist_font, Inline* }

attlist_font &=

attribute name { text }?,

attribute size { text }?,

attribute base { "default" | "last" }?,

attribute color_space { "RGB" }?,

attribute opacity { "100" }?,

attribute color { text }?,

attribute bold { Yes_No }?,

attribute underline { Yes_No }?,

attribute italic { Yes_No }?,

attribute oblique { Yes_No }?,

attribute small_caps { Yes_No }?,

attribute family {

"monospace" | "sans-serif" | "serif" | "cursive" | "fantasy"

}?,

attribute ul_type { "rightscore" | "leftscore" | "throughscore" }?,

attribute em_type {

"rightscore"

| "leftscore"

| "throughscore"

| "kendot"

| "bold"

| "italic"

| "bold_italic"

| "reverse"

| "shade"

}?

i = element i { attlist_i, Inline* }

attlist_i &= empty

u = element u { attlist_u, Inline* }

attlist_u &=

attribute ul_type { "rightscore" | "leftscore" | "throughscore" }?

em = element em { attlist_em, Inline* }

attlist_em &=

attribute em_type {

"rightscore"

| "leftscore"

| "throughscore"

| "kendot"

| "bold"

| "italic"

| "bold_italic"

| "reverse"

| "shade"

}?

oblq = element oblq { attlist_oblq, Inline* }

attlist_oblq &= empty

sc = element sc { attlist_sc, Inline* }

attlist_sc &= empty

mlg = element mlg { attlist_mlg, Inline* }

attlist_mlg &= empty

horizontal = element horizontal { attlist_horizontal, TextWithGaiji }

attlist_horizontal &= empty

ruby = element ruby { attlist_ruby, rbase, rtop }

attlist_ruby &= empty

```

rb = element rb { attlist_rb, TextWithGaiji }

attlist_rb &= empty

rt = element rt { attlist_rt, TextWithGaiji }

attlist_rt &= empty

sub = element sub { attlist_sub, TextWithGaiji }

attlist_sub &= empty

sup = element sup { attlist_sup, TextWithGaiji }

attlist_sup &= empty

external_char = element external_char { attlist_external_char, text }

attlist_external_char &=

    attribute alt_set { text }?,
    attribute alt_code { text }?,
    attribute alt_img { text }?,
    attribute alt_vimg { text }?,
    attribute img_type { text }?,
    attribute alt { text }?

mask =

    element mask {

        attlist_mask,

        (text | br | font | horizontal | ruby | external_char | object)*

    }

attlist_mask &=

    [ a:defaultValue = "on" ] attribute initial_flag { "on" | "off" }?,

    attribute trigger { text }?,

    attribute char_id { text }?,

    [ a:defaultValue = "default" ]

    attribute mask_type { "default" | "color" }?,

```

attribute color_space { "RGB" }?,
 attribute opacity { "100" }?,
 [a:defaultValue = "black"] attribute color { text }?,
 [a:defaultValue = "scope"] attribute hold_flag { HoldFlag }?
 char_id = element char_id { attlist_char_id, Inline* }
 attlist_char_id &= attribute char_id { text }

head = element head { attlist_head, headword+, key* }

attlist_head &= empty

headword = element headword { attlist_headword, TextWithGalji }

attlist_headword &=

attribute type { text }?,

attribute table_id { text },

[a:defaultValue = "IPA"] attribute phonetic_notation { text }?

key = element key { attlist_key, text }

attlist_key &= attribute type { text }?

meaning = element meaning { attlist_meaning, All_tag_p* }

attlist_meaning &=

attribute type { text }?,

attribute subid { text }?,

attribute level { text }?,

attribute no { text }?

example = element example { attlist_example, All_tag_p* }

attlist_example &= empty

subhead =

element subhead {

attlist_subhead, subheadword+, meaning*, example*, key*

}

attlist_subhead &=

attribute type { text }?,

attribute subid { text }?

subheadword = element subheadword { attlist_subheadword, TextWithGaiji }

attlist_subheadword &=

attribute type { text }?,

attribute subid { text }?

split = element split { attlist_split, All_tag_p* }

attlist_split &= attribute level { text }?

column = element column { attlist_column, All_tag_p* }

attlist_column &=

attribute type { text }?,

attribute subid { text }

object = element object { attlist_object, permission_info? }

attlist_object &=

attribute type { text },

attribute src { text },

attribute char_id { text }?,

attribute align { "top" | "middle" | "bottom" | "left" | "right" }?,

attribute start { "auto" | "event" }?,

attribute loop { "1" }?

\div = element div { attlist_div, All_tag_p* }

attlist_div &=

attribute level { text }?,

attribute title { text }?

span = element span { attlist_span, All_tag_p* }

attlist_span &=

attribute type { text }?,

attribute lang { text }?,

[a:defaultValue = "ISO 639-3"] attribute lang_code { text }?

table = element table { attlist_table, tr+ }

attlist_table &= empty

tr = element tr { attlist_tr, td+, th* }

attlist_tr &= empty

th = element th { attlist_th, Inline* }

attlist_th &= empty

td = element td { attlist_td, Inline* }

attlist_td &= empty

select = element select { attlist_select, select_item+ }

attlist_select &= empty

select_item = element select_item { attlist_select_item, All_tag_p* }

attlist_select_item &=

attribute default { text }?,

attribute type { text }?

tts = element tts { attlist_tts, All_tag_p* }

attlist_tts &= attribute speaker { "male" | "female" | "child" }?

a = element a { attlist_a, Inline* }

attlist_a &= attribute href { text }

dict_data =

element dict_data {

```
attlist_dict_data, dict_default_attribute, dict_body
}
```

```
attlist_dict_data &= empty
```

```
dict_default_attribute =
```

```
element dict_default_attribute {
    attlist_dict_default_attribute,
    dict_default_font?,
    dict_default_background?,
    dict_default_background_music?,
    line_breaking_method?
}
```

```
attlist_dict_default_attribute &=
```

```
attribute baseline { Baseline }?,
attribute valign { "middle" }?
```

```
dict_default_font =
```

```
element dict_default_font { attlist_dict_default_font, empty }
```

```
attlist_dict_default_font &=
```

```
attribute fontname { text }?,
attribute color_space { "RGB" }?,
attribute opacity { "100" }?,
```

```
attribute color { text }?,
```

```
attribute ruby_flag { "yes" | "yes_only" | "no" | "no_only" }?
```

```
dict_default_background =
```

```
element dict_default_background {
    attlist_dict_default_background, permission_info?
}
```

attlist_dict_default_background &=

attribute color_space { "RGB" }?,

attribute opacity { "100" }?,

attribute color { text }?,

attribute type { text }?,

attribute src { text }?

dict_default_background_music =

element dict_default_background_music {

attlist_dict_default_background_music, permission_info?

}

attlist_dict_default_background_music &=

attribute type { text },

attribute src { text },

[a:defaultValue = "no"] attribute loop { Yes_No }?

dict_body = element dict_body { attlist_dict_body, split*, dic-item+ }

attlist_dict_body &= empty

dic-item =

element dic-item {

attlist_dic-item,

(All_tag_di

| gender

| psp

| glabel

| pronunciation

| inflec

```

| slabel

| guideword

| spellout

| variant

| etymology)*

}

attlist_dic-item &=

    attribute type { text }?,

    attribute id { text },

    attribute rank { text }?,

    attribute level { text }?,

    [ a:defaultValue = "no" ] attribute page_break { Yes_No }?,

    attribute turning_page_control { Turn_Page_Val }?,

    attribute revision { text }?,

    attribute delete { Yes_No }?

gender = element gender { attlist_gender, Inline* }

attlist_gender &= empty

psp = element psp { attlist_psp, Inline* }

attlist_psp &= empty

glabel = element glabel { attlist_glabel, Inline* }

attlist_glabel &= empty

pronunciation = element pronunciation { attlist_pronunciation, Inline* }

attlist_pronunciation &=

    [ a:defaultValue = "IPA" ] attribute phonetic_notation { text }?

inflec = element inflec { attlist_inflec, Inline* }

attlist_inflec &= empty

slabel = element slabel { attlist_slabel, Inline* }

```

attlist_slabel &= empty

guideword = element guideword { attlist_guideword, Inline* }

attlist_guideword &= empty

spellout = element spellout { attlist_spellout, Inline* }

attlist_spellout &= attribute org { text }?

variant = element variant { attlist_variant, Inline* }

attlist_variant &= empty

etymology = element etymology { attlist_etymology, Inline* }

attlist_etymology &= empty

flip_animation =

element flip_animation {

attlist_flip_animation,

flip_animation_sound?,

flip_animation_source+

}

attlist_flip_animation &=

[a:defaultValue = "1s"] attribute renewal_time { text }?

flip_animation_sound =

element flip_animation_sound { attlist_flip_animation_sound, empty }

attlist_flip_animation_sound &=

attribute type { text },

attribute src { text }

flip_animation_source =

element flip_animation_source { attlist_flip_animation_source, empty }

attlist_flip_animation_source &=

attribute type { text },

```
attribute src { text },
attribute renewal_time { text }?
```

```
search_page =
```

```
element search_page {
    attlist_search_page,
    search_page_title?,
    key_input_region,
    key_input_region?,
    search_link_item*
}
```

```
attlist_search_page &= empty
```

```
search_page_title =
```

```
element search_page_title { attlist_search_page_title, TextWithGaiji }
```

```
attlist_search_page_title &=
```

```
attribute type { text }?,
attribute src { text }?
```

```
key_input_region =
```

```
element key_input_region {
    attlist_key_input_region, key_input_region_prompt, enable_key_type
}
```

```
attlist_key_input_region &=
```

```
attribute table_id { text },
attribute search_type { "matches_only" | "matches_first" }?
```

```
key_input_region_prompt =
```

```
element key_input_region_prompt {
    attlist_key_input_region_prompt, TextWithGaiji
```

```
}
```

```
attlist_key_input_region_prompt &= empty
```

```
search_link_item =
```

```
  element search_link_item {
```

```
    attlist_search_link_item, search_link_title
```

```
  }
```

```
attlist_search_link_item &= attribute char_id { text }
```

```
search_link_title =
```

```
  element search_link_title { attlist_search_link_title, TextWithGaji }
```

```
attlist_search_link_title &= empty
```

```
rbase |= notAllowed
```

```
rtop |= notAllowed
```

```
start =
```

```
  text_data
```

```
  | bvf
```

```
  | rt
```

```
  | rb
```

```
  | flow_default_delimiter
```

```
  | dict_data
```

```
  | search_page
```

```
  | flip_animation
```

Bibliography

The following documents have served as references in the preparation of this International Standard:

ISO/IEC 9541-1:1991, *Information technology – Font information interchange – Part 1 Architecture*

ISO/IEC 10179:1996, *Information technology – Processing languages – Document Style Semantics and Specification Language (DSSSL)*

ISO 639-3:2007, *Codes for the representation of names of languages – Part 3: Alpha-3 code for comprehensive coverage of languages*

ISO 646, *Information technology – ISO 7-bit coded character set for information interchange*

ISO 3166-1, *Codes for the representation of names of countries and their subdivisions – Part 1: Country codes*

ISO 8601, *Data elements and interchange formats – Information interchange – Representation of dates and times*

ISO 8859-15, *Information technology – 8-bit single-byte coded graphic character sets – Part 15: Latin alphabet No. 9*

ISO 8879:1986, *Information processing – Text and office systems – Standard Generalized Markup Language (SGML)*

Corrigendum 2 (1999)

W3C Recommendation, *Extensible Markup Language (XML) 1.0* (Third Edition), 2004-02, <http://www.w3.org/TR/2004/REC-xml-20040204>

W3C Recommendation, *Extensible Stylesheet Language (XSL) Version 1.0*, 2001-10, <http://www.w3.org/TR/2001/REC-xsl-20011015/>

Internet Engineering Task Force(IETF), "*RFC 3986-Uniform Resource Identifier (URI): Generic Syntax*", <http://www.ietf.org/rfc/rfc3986.txt>

IEC 62448:2009, *Multimedia systems and equipment – Multimedia E-publishing and E-books – Generic format for E-publishing*

Digital ASSIST Ltd., *Digital ASSIST*, <http://www.d-assist.com/index.html>

T. Berners-Lee et al., RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax*, January 2005 <http://tools.ietf.org/html/rfc3986>

JIS X 0208:1997, *7-bit and 8-bit double byte coded KANJI sets for information interchange*, Japan Standards Association, January 1997
Available in Japanese only.

JIS X 0213:2000/AMENDMENT 1:2004, *7-bit and 8-bit double byte coded extended KANJI sets for information interchange*, Japan Standards Association, February 2004
Available in Japanese only.

IECNORM.COM: Click to view the full PDF of IEC 62605:2011

SOMMAIRE

AVANT-PROPOS	131
INTRODUCTION.....	133
1 Domaine d'application	134
2 Références normatives.....	134
3 Termes, définitions et index des termes utilisés.....	134
4 Position et exigences relatives au format d'échange des dictionnaires électroniques.....	135
4.1 Format d'échange pour dictionnaires électroniques dans le modèle de création/distribution de contenus	135
4.2 Exigences relatives au format d'échange des dictionnaires électroniques.....	136
5 Format de fichier	137
6 Sémantique	137
7 Caractéristiques du format de fichier	137
8 Présentation de la structure du format	137
9 Éléments et attributs.....	138
9.1 Généralités.....	138
9.2 Page_ID	138
9.3 Object_ID	139
9.4 Char_ID	139
9.5 Reading	139
9.6 Filename Nom de fichier.....	140
9.7 Caractère standard.....	140
9.8 Chaîne de caractères standard.....	140
9.9 Caractère étendu.....	140
9.10 Chaîne de caractères étendus.....	141
9.11 Caractère externe	141
9.12 Chaîne de caractères externes.....	142
9.13 Chaîne de caractères externes étendus	142
9.14 Coordonnées.....	143
9.15 Polygonal_region.....	143
9.16 Color.....	143
9.17 Date.....	144
9.18 Time.....	144
9.19 Country	144
9.20 Personal_name	144
9.21 "Organization_name".....	145
9.22 "Address"	145
9.23 "Permission"	146
9.24 "Keyword"	147
9.25 "Telephone_number"	147
9.26 "Mail_address"	148
10 Détails du format de description	149
10.1 Généralités.....	149
10.2 Modules d'information du livre <bvf>	149
10.2.1 Généralités.....	149
10.2.2 Données bibliographiques <book_info>	150

10.3	Module de gestion de contenu <body_module>	157
10.3.1	Généralités	157
10.3.2	Données de contenu courant <flow_type_body>	157
10.4	Module information-événement <event_info>	171
10.4.1	Événements	171
10.4.2	Données d'événement <event>	172
10.4.3	Déclencheur (pointeur) <trigger_pointer>	172
10.4.4	Informations d'action	174
10.5	Module composantes <parts_module>	176
10.5.1	Stockage et gestion	176
10.5.2	Objet texte dynamique <dynamic_text_object_entry>	177
10.5.3	Objet son <sound_object_entry>	178
10.5.4	Objet page de recherche <search_page_object_entry>	178
10.5.5	Objet film <movie_object_entry>	179
10.5.6	Objet données de dictionnaire <dict_data_object_entry>	179
10.6	Instanciation d'un objet	180
10.6.1	Généralités	180
10.6.2	Instanciation d'objet texte <text_data>	180
10.6.3	Instanciation d'un objet données de dictionnaire	204
10.6.4	Instanciation d'un objet image	210
10.6.5	Instanciation d'un objet son	210
10.6.6	Instanciation d'un objet animation	210
10.6.7	Instanciation d'un objet page de recherche <search_page>	212
10.6.8	Instanciation d'objet film	216
Annexe A (normative)	Noms des couleurs	217
Bibliographie		259
Figure 1	– Modèle de création/distribution de contenus	136
Figure 2	– Modèle de création/distribution de contenus (modifié)	136
Figure 3	– Relation entre les concepts	136
Figure 4	– Structure de l'arborescence XML	138
Figure 5	– Exemple de valign="middle"	182
Figure 6	– Exemple de lettrine	186
Figure 7	– Marges de gauche et de droite d'un paragraphe	186
Figure 8	– Écriture horizontale dans un texte vertical	192
Figure 9	– Ruby	193
Figure 10	– Exemple de rendu d'une instanciation d'objet page de recherche	213
Tableau 1	– Caractères de base pour "Reading" ou "lecture"	139
Tableau 2	– Jeu de caractères standard	140
Tableau 3	– Caractères utilisables pour un numéro de téléphone	148
Tableau 4	– Caractères utilisables pour des adresses électroniques	148
Tableau 5	– Caractères utilisables pour la clé de recherche	170
Tableau A.1	– Noms de couleur et à titre de référence terme anglais correspondant	217
Tableau A.2	– Exemples de jeux supplémentaires de caractères standard	218

Tableau A.3 – Exemple de caractères supplémentaires utilisables pour des transcriptions	219
Tableau A.4 – Exemple de règles de tri supplémentaires	220
Tableau A.5 – Exemple d'attributs supplémentaires spécifiques à une langue pour <enable_key_type>	221

IECNORM.COM: Click to view the full PDF of IEC 62605:2011

Withdrawing

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

SYSTÈMES ET ÉQUIPEMENTS MULTIMÉDIA – PUBLICATION ET LIVRES ÉLECTRONIQUES MULTIMÉDIA – FORMAT D'ÉCHANGE POUR LES DICTIONNAIRES ÉLECTRONIQUES

AVANT-PROPOS

- 1) La Commission Electrotechnique Internationale (CEI) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de la CEI). La CEI a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, la CEI – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de la CEI"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec la CEI, participent également aux travaux. La CEI collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de la CEI concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de la CEI intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de la CEI se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de la CEI. Tous les efforts raisonnables sont entrepris afin que la CEI s'assure de l'exactitude du contenu technique de ses publications; la CEI ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de la CEI s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de la CEI dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de la CEI et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) La CEI elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de la CEI. La CEI n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à la CEI, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de la CEI, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de la CEI ou de toute autre Publication de la CEI, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.

La Norme Internationale CEI 62605 a été établie par le domaine technique 10: Multimedia e-publishing and e-book¹ (publication et livre électronique multimédia), du comité d'études 100 de la CEI: Systèmes et appareils audio, vidéo et multimédia.

Le texte de cette norme est issu des documents suivants:

FDIS	Rapport de vote
100/1829/FDIS	100/1863/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cette norme.

Cette publication a été rédigée selon les Directives ISO/CEI, Partie 2.

¹ Publication et livre électronique multimédia. Le titre du domaine technique 10 existe en anglais seulement.

Le comité a décidé que le contenu de cette publication ne sera pas modifié avant la date de stabilité indiquée sur le site web de la CEI sous "<http://webstore.iec.ch>" dans les données relatives à la publication recherchée. A cette date, la publication sera

- reconduite,
- supprimée,
- remplacée par une édition révisée, ou
- amendée.

IEC NORM.COM: Click to view the full PDF of IEC 62605:2011

Withdrawn

INTRODUCTION

Les marchés des livres et publications électroniques multimédia appellent la normalisation des formats d'échange de données électroniques entre les personnes concernées, auteurs, préparateurs de données, éditeurs et lecteurs. Les formats sont classés en format de soumission, format d'échange et format de lecture. Le format de soumission assure une interaction entre les auteurs et les préparateurs de données. Le format de lecture dépend du matériel de e-publishing (terminal de lecture ou publication électronique). Le format d'échange fournit un format d'échange aux préparateurs de données et aux éditeurs, et il convient en conséquence qu'il soit indépendant du matériel de publication électronique.

La Commission Électrotechnique Internationale (CEI) attire l'attention sur le fait qu'il est déclaré que la conformité aux dispositions du présent document peut impliquer l'utilisation d'un brevet.

La CEI ne prend pas position sur la preuve, la validité et le domaine d'application de ce droit de propriété intellectuelle.

Sharp Corporation a informé la CEI qu'elle possède des droits de propriété ou a octroyé des droits de propriété.

Le détenteur de ce droit de propriété a assuré à la CEI qu'il est prêt à négocier des licences avec les demandeurs dans le monde entier, dans des conditions raisonnables et non discriminatoires. A cet égard, la déclaration du détenteur de ce droit de propriété est enregistrée auprès de la CEI. Des informations peuvent être obtenues auprès de:

Sharp Corporation
Platform Technology Development Center
2613-1, Ichinomoto-cho, Tenri, Nara
632-8587 JAPON

L'attention est attirée sur le fait que certains des éléments du présent document peuvent faire l'objet de droits de propriété autres que ceux identifiés ci-dessus. La CEI ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de propriété et de ne pas avoir signalé leur existence.

L'ISO (www.iso.org/patents) et la CEI (<http://patents.iec.ch/>) tiennent à jour des bases de données en ligne de brevets correspondants à leurs normes. Les utilisateurs souhaitant obtenir les dernières informations concernant les brevets sont invités à consulter les bases de données.

SYSTÈMES ET ÉQUIPEMENTS MULTIMÉDIA – PUBLICATION ET LIVRES ÉLECTRONIQUES MULTIMÉDIA – FORMAT D'ÉCHANGE POUR LES DICTIONNAIRES ÉLECTRONIQUES

1 Domaine d'application

La présente Norme Internationale spécifie un format d'échange de dictionnaires électroniques entre les éditeurs, les créateurs de contenu et les fabricants.

La présente norme internationale ne traite pas des aspects suivants:

- les formats de données des appareils de lecture;
- les éléments nécessaires uniquement à l'impression finale;
- les questions de rendu liées aux dispositifs physiques;
- les questions de sécurité (la gestion des droits numériques d'un document, par exemple).

2 Références normatives

Les documents de référence suivants sont indispensables pour l'application du présent document. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

CEI/TS 62229:2006, *Multimedia systems and equipment – Multimedia e-publishing and e-book – Conceptual model for multimedia e-publishing*

Disponible en anglais seulement.

3 Termes, définitions et index des termes utilisés

Pour les besoins du présent document, les termes et définitions suivants s'appliquent.

3.1

fabricant

organisation ou personne qui fabrique le matériel et/ou le logiciel du livre électronique

3.2 Index des termes utilisés

Français	Anglais
article	entry
autorisations	permissions
balise d'autorisation	permission tag
balise de racine	root tag
caractère étendu	extended character
chaîne	string
chaîne de caractères étendus	extended character string
chaînes de mots indexés	entry word strings
chaîne de caractères standard	standard character string
chemin	path
contenu courant	flowing contents
correspondance (exacte) uniquement	matches-only
données d'événement	event data
données d'un article	entry data
flot de texte	text flow
foobar dictionnaire	FooBar dictionary
image hypertexte	clickable image map
instanciation de l'objet	object instance
instanciation d'objet données de dictionnaire	dictionary data object instance
instanciation d'objet texte	text object instance
jeu de caractères	charset
lecture	playback
lecture d'un mot	reading
livret	wordbook
mise automatique en signet	automatic bookmarking
mots indexés	entry words
nom de fichier	filename
paragraphe	article
plage de paires de substitution	surrogate pair range
première correspondance	matches-first
sous-chaînes	substrings
terminal de lecture / publication électronique	e-publishing
utilisateur authentifié	authenticated user
zones	regions

4 Position et exigences relatives au format d'échange des dictionnaires électroniques

4.1 Format d'échange pour dictionnaires électroniques dans le modèle de création/distribution de contenus

Le modèle conceptuel de publication électronique multimédia (CEI/TS 62229) définit un modèle de création/distribution de contenus montré à la Figure 1.

Auteur <--(1)--> Préparateur de données <--(2)--> Éditeur --(3)--> Lecteur

IEC 1383/11

Légende

- (1) données de contenu au format de soumission
- (2) données de contenu au format d'échange
- (3) données de contenu au format de lecture

Figure 1 – Modèle de création/distribution de contenus

Il convient de noter que le rôle des fabricants de matériel ou de logiciel de dictionnaire électronique empiète sur celui de l'éditeur dans la Figure 1. Par conséquent, un modèle légèrement modifié est spécifié dans la présente norme tel que présenté dans la Figure 2.

Auteur <--(1)--> Préparateur de données <--(2)--> Éditeur (fabricant) --(3)--> Lecteur

IEC 1385/11

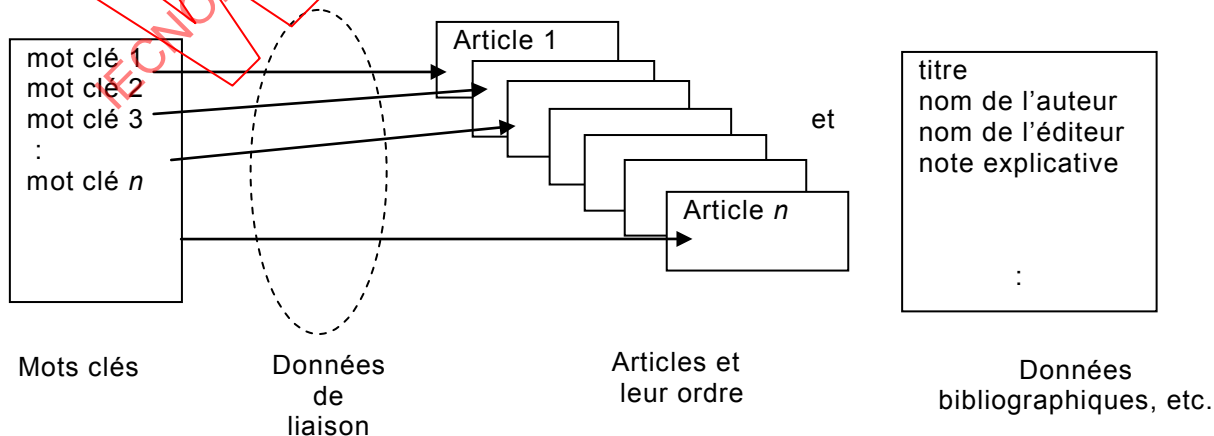
Figure 2 – Modèle de création/distribution de contenus (modifié)

La présente norme spécifie le format d'échange entre les préparateurs de données et les éditeurs, c'est-à-dire un format pour le lien (2) dans la Figure 2, même s'il peut être utilisé comme un format de lecture.

4.2 Exigences relatives au format d'échange des dictionnaires électroniques

Un format d'échange pour dictionnaires électroniques doit traiter des points suivants.

- Description des mots-clés, liens allant des mots-clés aux articles (en anglais *entry*) , données de liaison) et ordre des articles.
- Description des paragraphes (en anglais *articles*) de chaque article (données d'un article ou *entry data*). Cela comprend texte, image et autres fonctionnalités multimédia dont ont généralement besoin les livres électroniques.
- Description des données bibliographiques et autres données. Il convient d'inclure le nom de l'auteur et de l'éditeur, le titre du contenu et la note explicative. La relation entre ces concepts est représentée visuellement dans la Figure 3.
- Description du contenu rédigée en différentes langues.



IEC 1385/11

Figure 3 – Relation entre les concepts

5 Format de fichier

La présente norme internationale repose sur les formats XMDF (décrit dans la CEI 62448:2009, Annexe B) et LeXML. Le format standard est représenté en XML et est ici appelé format XMDF-LeXML.

NOTE LeXML est proposé par Digital ASSIST Ltd. Les spécifications d'origine peuvent être consultées à <http://www.d-assist.com/index.html>.

6 Sémantique

Le rendu d'éléments au format XMDF-LeXML est possible avec des spécifications de style appropriées qui sont en dehors du domaine de la présente norme internationale.

7 Caractéristiques du format de fichier

Le présent article spécifie le format XMDF-LeXML mentionné à l'Article 5.

Le format XMDF-LeXML est un format d'échange de données pour dictionnaires électroniques de type livre électronique multimédia; ce format est destiné aux préparateurs de données et aux éditeurs plutôt qu'aux lecteurs, et l'accent y est mis sur les dispositifs mobiles, qui sont la plate-forme ciblée. À l'instar de HTML, ce format ne divise pas le document en pages fixes, mais détermine la mise en page en fonction de la taille d'écran du dispositif de visualisation, de la police utilisée, etc. Dans la présente norme, ce type de contenu est appelé contenu courant, par opposition à du contenu paginé.

8 Présentation de la structure du format

Le contenu courant est en général composé de plusieurs flots concaténés. La présente norme ne précise aucune exigence particulière sur la façon de diviser le contenu courant en flots individuels. Cette décision revient au préparateur de données, pour traiter différents types de contenus. Par exemple, un journal peut comporter un flot par article, un roman un flot par chapitre, et ainsi de suite. Il est également possible de ne pas diviser le contenu et de n'avoir qu'un seul flot. Toutefois, il convient de noter que des flots particulièrement volumineux, ou un très grand nombre de flots, peuvent avoir un effet sur le temps de traitement selon la version spécifique du dispositif de visualisation utilisé, la mémoire disponible, etc.

La structure de l'arborescence XML du format est montrée à la Figure 4.

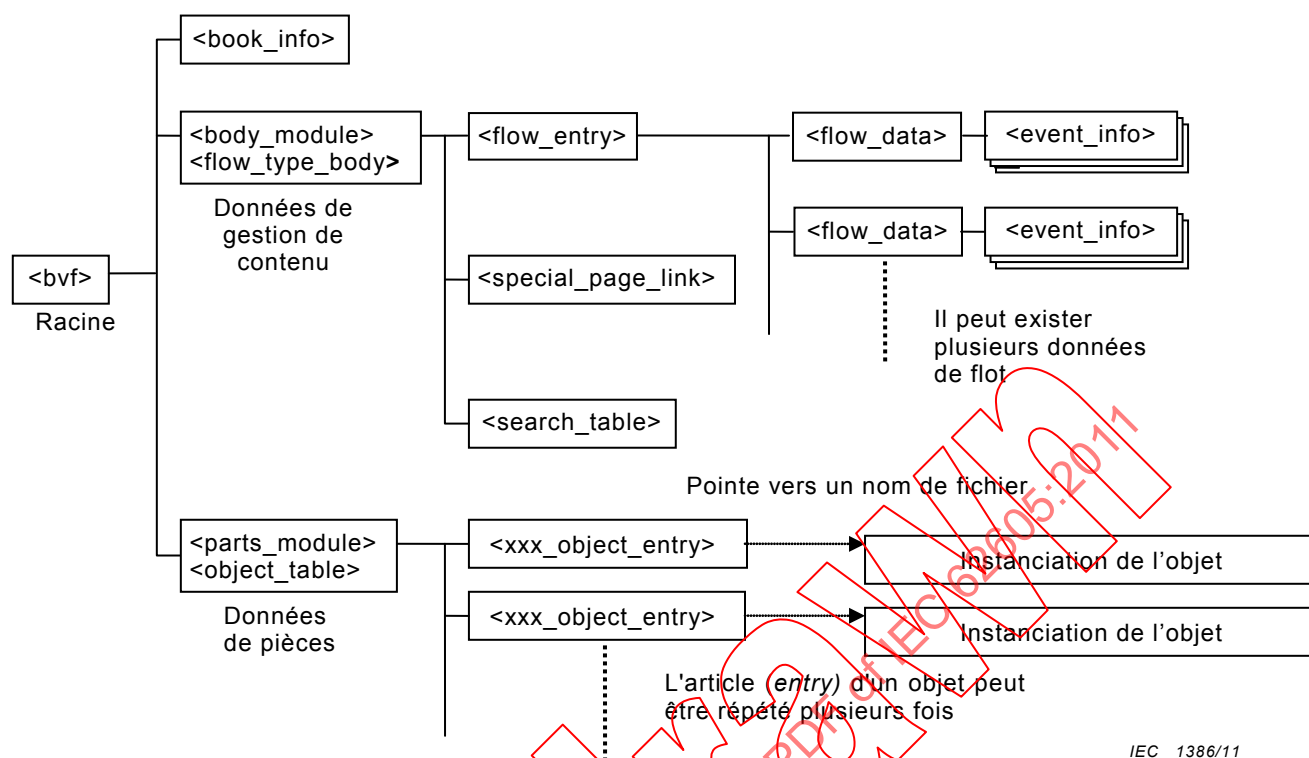


Figure 4 – Structure de l'arborescence XML

Le contenu réel de chaque flot, ou en d'autres termes ce que le dispositif de visualisation affiche, est enregistré dans l'*instanciation de l'objet*. L'*instanciation de l'objet* est enregistrée dans *object_entry* et associée à un ID et à d'autres données auxiliaires permettant d'en faire des données lisibles/affichables. *Flow_data* détermine son contenu en pointant vers ces objets enregistrés. De plus, les informations relatives aux fonctionnalités (un lien de page, par exemple) sont enregistrées dans *event_info*.

La partie principale de la présente norme est générique et peut être utilisée dans tous les pays et toutes les langues. Toutefois, certaines parties peuvent avoir un comportement spécifique à la langue. Pour que la partie principale de cette spécification reste lisible et bien ciblée, les questions liées à la localisation linguistique sont présentées en détail dans l'article A.2. Le texte principal y fera référence en cas de besoin.

9 Éléments et attributs

9.1 Généralités

Les différents types de valeurs qui peuvent être utilisés dans les balises ou attributs sont décrits ci-dessous. Les éléments et attributs détaillés ci-après sont valides tout au long de la présente norme et seront employés par d'autres constructions. Dans les explications suivantes, les caractères alphanumériques font référence aux nombres compris entre 0 et 9 et aux caractères de l'alphabet de a à z et A à Z.

9.2 Page_ID

Page_ID spécifie un numéro d'identification unique pour les données de flot du contenu courant. Il s'agit d'une chaîne commençant par les caractères "PG" suivis de caractères alphanumériques.

Exemple:

```
<flow_data flow_id="PG0002" ... />
```

9.3 Object_ID

Object_ID spécifie un numéro d'identification unique pour les objets utilisés dans le contenu courant. Il s'agit d'une chaîne commençant par les caractères "OB" suivis de caractères alphanumériques.

Exemple:

```
<dynamic_text_object_entry id="OB0ue4".../>
```

9.4 Char_ID

Char_ID spécifie un numéro d'identification pour les positions (chaînes de caractères, etc.) à l'intérieur d'objets de texte et d'objets de données de dictionnaire. Il s'agit d'une chaîne alphanumérique qui doit apparaître de manière unique dans le *texte* (voir 10.6.2) et l'*instanciation d'objet données de dictionnaire* (voir 10.6.3). Les parties Char_ID portant la même valeur dans différentes instanciations d'objet sont considérées comme distinctes et n'ont aucun impact l'une sur l'autre.

Exemple:

```
<trigger_pointer id="OB29s0/CR0de4"/>
Click<char_id char_id="CR0001">here</char_id>for details.
```

9.5 Reading

Pour faciliter le tri, la donnée "Reading" ou "lecture" peut être utile pour spécifier la lecture de chaque mot. Limiter la taille de l'ensemble des caractères permis dans "Reading" ou "lecture" simplifie le tri. Les caractères de cet "ensemble des caractères permis" sont à déterminer pour chaque langue. Toutes les langues peuvent utiliser les caractères énumérés au Tableau 1 comme base commune, puisque la localisation (voir Article A.2) décrira les extensions spécifiques à une langue.

Tableau 1 – Caractères de base pour "Reading" ou "lecture"

Nom	Caractères correspondants ^{a)}
Alphabet de base	A à Z (0x0041 à 0x005A) A à z (0x0061 à 0x007A)
Nombres	0 à 9 (0x0030 à 0x0039)
Autres	espace (0x0020), ((0x0028),) (0x0029)
a) Toutes les valeurs sont au format Unicode.	

Exemple:

```
<title reading="PI">π</title>
```

9.6 Filename Nom de fichier

Les noms de fichier doivent être écrits selon la convention ci-dessous. Le chemin se rapporte au fichier dans lequel cette référence est faite. Il convient de ne pas utiliser de "chemins réseau". Pour des raisons de portabilité, il est recommandé de n'utiliser que des caractères ACSII. La barre oblique et la barre oblique inversée sont admises comme séparateurs de répertoire. Il est également recommandé de ne pas utiliser de noms de fichier trop longs. Ils risqueraient de ne pas être pris en charge par le système d'exploitation hôte.

Exemple:

```
<dynamic text object entry src="sect1.xml" type="text/x-bvf-text" id="OB0ue4"/>
```

9.7 Caractère standard

Le jeu de caractères standard du document, tel que défini par l'attribut `default_ccs` de la balise `<bvf>` (voir 10.2), doit être choisi dans une liste bien définie de manière à faciliter le développement du logiciel de visualisation. Toutefois, cette liste peut changer en fonction des différentes versions localisées du format XMDF-LeXML. Toutes les données de livre électronique doivent définir leur jeu de caractères standard comme l'un des jeux de caractères figurant dans le Tableau 2 ou une combinaison de tels jeux de caractères et ceux définis pour une localisation particulière (voir A.2.2).

Tableau 2 – Jeu de caractères standard

Nom du jeu de caractères	Description
"ISO 646-IRV"	Caractères US-ASCII

9.8 Chaîne de caractères standard

Une chaîne composée de *caractères standard* est appelée *chaîne de caractères standard*. Sauf indication contraire, les caractères d'espacement (espace (0x0020), les sauts de ligne (0x000D, 0x000A, 0x0D0A) et les tabulations (0x0009)) doivent être traités comme suit:

L'espace (0x0020) doit être affiché en l'état.

Le saut de ligne (0x000D, 0x000a, 0x0D0A) ne doit pas être affiché, mais simplement omis.

La tabulation (0x0009) doit être affichée comme s'il s'agissait d'un espace simple.

En outre, en raison d'une restriction dans le format XML, lors de la conversion au format de distribution les sauts de ligne (0x000D, 0x000A, 0x0D0A) et les tabulations (0x0009) trouvés dans les valeurs d'attribut sont à remplacer par des espaces.

9.9 Caractère étendu

Les caractères contenant des points de code Unicode mais ne faisant pas partie de ceux répertoriés ci-dessous sont appelés *caractères étendus*.

Caractères standard

Plage des paires de substitution (*Surrogate pair range*) (0xD800 à 0xDFFF)

BOM (Byte Order Mark) (0xFFFF, 0xFEFF)

NON CHARACTER (0xFFFF)

Les caractères de contrôle (caractères entre 0x0000 et 0x001F, à l'exception de la tabulation (0x0009) et du saut de ligne (0x000A, 0x000D), et DEL (0x007F))

Si un livre électronique utilise un *Caractère étendu* dans ses données, il convient que le nom d'un jeu de caractères couvrant ces *caractères étendus* soit ajouté à l'attribut *default_ccs* de la balise <bvf>. Noter que tous les *Caractères étendus* utilisés dans le document n'ont pas besoin d'être couverts par le même jeu de caractères, étant donné qu'il est possible d'en spécifier plusieurs.

9.10 Chaîne de caractères étendus

Une chaîne composée de *caractères standard* et de *caractères étendus* est appelée *chaîne de caractères étendus*. Sauf indication contraire, les caractères d'espacement (espace (0x0020), saut de ligne (0x000D, 0x000A, 0x0D0A), tabulation (0x0009)) doivent être traités de la même manière que dans les *chaînes de caractères standard*.

9.11 Caractère externe

Pour afficher un caractère qui n'est ni un *caractère standard* ni un *caractère étendu*, il est possible d'utiliser la balise <external_char> décrite ci-dessous.

<external_char> permet d'insérer un *caractère externe*. Le dispositif de visualisation peut l'afficher conformément aux méthodes suivantes.

- a) Affichage du jeu de caractères par les attributs alt_set et alt_code.
- b) Affichage de l'ensemble d'images par les attributs alt_img ou alt_vimg.
- c) Affichage de l'ensemble de lettres alternatif par l'attribut alt.

Ses attributs sont les suivants.

[Attributs]

alt_set: Avec l'attribut alt_code, il permet de désigner le *caractère externe* à utiliser. Cet attribut alt_set indique le nom de la police, alors que l'attribut alt_code indique le point de code du caractère à l'intérieur de la police. L'attribut alt_set est écrit de la manière suivante:

alt_set = "font1,font2,..."

L'attribut alt_set peut contenir plusieurs noms de police, séparés par ",", (0x002C). Dans ce cas, il convient que le dispositif de visualisation utilise la première police disponible de la liste (présente sur la plate-forme ou incluse dans le contenu lui-même) pour afficher ce caractère.

alt_code: Permet de sélectionner un point de code de caractère dans la police spécifiée par l'attribut alt_set. Il peut être écrit sous la forme d'un nombre décimal ou hexadécimal, précédé du préfixe "0x". Si plusieurs polices ont été définies dans l'attribut alt_set, le code de caractère doit représenter le même caractère dans chacune d'elles. Cet attribut peut être omis.

alt_img: Permet de définir un autre dessin de caractère. Écrit sous la forme d'un *nom de fichier*. Avant d'ouvrir le fichier indiqué par cet attribut, il convient de vérifier l'attribut img_type pour connaître les types de fichier autorisés. Noter qu'il peut être utilisé uniquement lorsque <external_char> est utilisé dans une *instanciation d'objet texte*. Si alt_img et alt_vimg sont tous deux utilisés, les types de fichier doivent correspondre. Cet attribut peut être omis. Si une valeur est affectée à cet attribut, il convient que l'affichage soit conforme aux méthodes suivantes:

- a) Images monochromes

Les pixels noirs et blancs représentent respectivement la lettre et l'arrière-plan. Les couleurs de police et d'arrière-plan doivent s'afficher conformément à l'attribut de couleur de la balise .

- b) Images avec niveaux de gris

Les pixels noirs et blancs représentent respectivement la lettre et l'arrière-plan. Les couleurs de police et d'arrière-plan doivent s'afficher conformément à l'attribut de couleur de la balise . La couleur des pixels "gris" doit être calculée comme une valeur intermédiaire entre la couleur de police et la couleur d'arrière-plan.

c) Images couleur

Affichées en l'état.

alt_vimg: Permet de définir un autre dessin du caractère à utiliser lorsque le texte s'affiche verticalement (comme cela peut être le cas dans certaines langues, comme le japonais, par exemple). Écrit sous la forme d'un *nom de fichier*. S'il est omis, il convient d'utiliser l'image définie dans l'attribut alt_img et pour la présentation horizontale et pour la présentation verticale. Avant d'ouvrir le fichier indiqué par cet attribut, il convient de vérifier l'attribut img_type pour savoir si c'est un type de fichier autorisé. Noter qu'il peut être utilisé uniquement lorsque <external_char> est utilisé dans une *instanciation d'objet texte*. Si alt_img et alt_vimg sont utilisés tous deux, les types de fichier doivent correspondre. Cet attribut peut être omis. Si une valeur est affectée à cet attribut, il convient que l'affichage se fasse avec les mêmes méthodes que pour alt_img.

img_type: Permet de définir le type mime des fichiers d'images définis dans les attributs alt_img et alt_vimg. Actuellement, seuls PNG et JPEG sont pris en charge, et il convient de les écrire de la manière suivante:

"image/png"

"image/jpeg"

Si une valeur est affectée à alt_img et/ou alt_vimg, cet attribut est obligatoire. Comme ces deux attributs, il peut uniquement être utilisé dans une balise <external_char> à l'intérieur d'une *instanciation d'objet texte*.

alt: Autre chaîne de caractères. Écrit comme une *chaîne de caractères standard*. Peut être omis.

Exemple:

```
<external_char alt_set="oooextchars" alt_code="47268" alt_img="ou.img"
alt_vimg="ou_v.img" img_type="image/jpeg" alt="鷗"/>
<external_char alt="間"/>
<external_char alt_set="sharp_extchars" alt_code="0x2345" alt="間"/>
```

9.12 Chaîne de caractères externes

Une *chaîne de caractères externes* est une chaîne composée de *caractères standard*, de *caractères externes* ou des deux. Sauf indication contraire, il convient de traiter les caractères d'espacement (espace (0x0020), saut de ligne (0x000D, 0x000A) et tabulation (0x0009)) de la même manière que dans les *chaînes de caractères standard*.

Exemple:

```
森<external_char alt_img="ou.png" alt_type="image/png" alt="鷗"/>外
内田百 <external_char alt_set="sharp_extchars" alt_code="0x2345" alt="間"/>
```

9.13 Chaîne de caractères externes étendus

Une *chaîne de caractères externes étendus* est une chaîne composée de *caractères standard*, *caractères étendus*, *caractères externes* ou d'une *combinaison quelconque de ceux-ci*. Sauf indication contraire, il convient de traiter les caractères d'espacement (espace (0x0020), saut de ligne (0x000D, 0x000A) et tabulation (0x0009)) de la même manière que dans les *chaînes de caractères standard*.

9.14 Coordonnées

Type de données permettant de stocker des coordonnées, des dimensions et d'autres informations analogues composées des valeurs d'un x et d'un y. Elles se présentent sous la forme "(x, y)". Le nom de l'attribut qui utilise ce type de données dépend de la balise.

Le système de coordonnées décrit ci-dessous est utilisé dans la présente norme. L'origine se trouve dans le coin supérieur gauche, l'axe des abscisses x étant orientée vers la droite et l'axe des ordonnées y vers le bas. Étant donné que le système de coordonnées utilisé par le dispositif de visualisation pour mettre des éléments en correspondance à l'écran dépend de la mise en œuvre, il n'est pas présenté ici.

Système de coordonnées local:

L'origine du système de coordonnées local d'un objet se trouve dans le coin supérieur gauche du rectangle circonscrit, son orientation étant identique à celle du système de coordonnées général. Il convient d'exprimer les positions à l'intérieur d'un objet dans le système de coordonnées local.

Exemple:

```
<vertex position="(100,200)"/>
```

9.15 Polygonal_region

Format de données permettant de stocker les sommets d'un polygone ou toute autre suite ordonnée des sommets. Chaque sommet est stocké dans une balise <vertex>. Lors de la définition de la forme du polygone, les arêtes ne doivent pas se croiser. Si c'est le cas, le comportement du dispositif de visualisation n'est pas spécifié. La balise <vertex> comporte l'attribut ci-dessous.

[Attribut]

position: position du sommet, exprimée par des *coordonnées*. Cet attribut ne doit pas être omis.

Exemple:

```
<vertex position="(100,0)"/> <!--dans le cas d'un triangle -->
<vertex position="(0,100)"/>
<vertex position="(200,100)"/>
```

9.16 Color

Type de données de définition des couleurs. Les attributs ci-dessous sont définis.

[Attributs]

color_space: Permet de spécifier l'espace de couleur à utiliser. Actuellement, seul RGB est accepté. Si cet attribut est omis, il convient que le dispositif de visualisation agisse comme si RGB était indiqué.

color: Permet d'indiquer le nom de la couleur. Les noms de couleur ou les valeurs numériques peuvent être utilisés. Les noms de couleur admis sont répertoriés dans le Tableau A.1. La valeur par défaut dépend de la balise et du contexte réels. Les valeurs numériques doivent être écrites en respectant la syntaxe suivante.

Dans RGB: écrites sous la forme #RRGGBB. RR, GG, BB étant des nombres hexadécimaux, entre 00 et FF. Les valeurs de pixel d'échelle de gris sont représentées en attribuant la même valeur à RR, GG et BB.

opacity: niveau d'opacité. Compris entre 0 (transparent) et 100 (opaque). Actuellement, la seule valeur admise est 100, et si l'attribut est omis, sa valeur par défaut est 100.

Exemple:

```
<font color="#FF0000"/>
    <!-- le color_space n'est pas spécifié et prend la valeur par
    défaut RGB-->
<font color="#FF0000" opacity="100"/>
<font color="black"/>
```

9.17 Date

Format de données permettant de stocker des dates. Il utilise la même représentation que l'ISO 8601. Par exemple, 1994-11-05T08:15:30-05:00 correspond au 5 novembre 1994, 8:15:30, Heure normale de l'Est des Etats-Unis. Les formes abrégées sont également admises. Voir <http://www.w3.org/TR/NOTE-datetime> pour plus de détails.

Exemple:

```
<publication date type="publish">1994</publication date>
```

9.18 Time

Format de données permettant de préciser les durées. Écrit sous la forme "XXdXXhXXmXXsXXXms", où X est un chiffre compris entre 0 et 9. Par exemple, "10d5h30m10s015ms" signifie 10 jours, 5 heures, 30 minutes, 10 secondes et 15 millisecondes. Les formes abrégées ("5m30ms" ou "1s", par exemple) sont possibles. Le nombre de jours est illimité.

Exemple:

```
<flip animation renewal time="1s" >
```

9.19 Country

Format de données permettant de spécifier le nom d'un pays. Écrit conformément à l'ISO 3166-1 alpha 3, en minuscules.

Exemple:

```
<publication_place>jpn</publication_place>
```

9.20 Personal_name

Format de données permettant de stocker des noms de personnes (l'auteur du document, par exemple). Les noms sont placés dans la balise <personal_name>. On utilise plusieurs "éléments-enfants" pour définir les différentes parties du nom: le prénom, le deuxième prénom et le nom de famille. Ces informations doivent être inscrites par l'intermédiaire des balises définies ci-dessous. L'un au moins de <first_name>, <middle_name> et <last_name> doit être spécifié.

[Éléments-enfants]

<first_name> Permet de définir le prénom. Écrit comme une *chaîne de caractères externes*. Peut être omis. Accepte l'attribut suivant:

[Attribut]

reading: Prononciation du prénom, écrit comme un attribut *Reading*. Peut être omis.

<middle_name> Permet de définir le deuxième prénom. Écrit comme une *chaîne de caractères externes*. Peut être omis. Accepte l'attribut suivant:

[Attribut]

reading: Prononciation du deuxième prénom, écrit comme un attribut *Reading*. Peut être omis.

<last_name> Permet de définir le nom de famille. Écrit comme une *chaîne de caractères externes*. Peut être omis. Accepte l'attribut suivant:

[Attribut]

reading: Prononciation du nom de famille, écrit comme un attribut *Reading*. Peut être omis.

Exemple:

```
<personal_name>
  <!-- pour enregistrer John Smith -->
  <last_name>Smith</last_name>
  <first_name>John</first_name>
</personal_name>
```

9.21 "Organization_name"

Format de données permettant de définir le nom de l'entreprise (les éditeurs, par exemple). Il est placé dans la balise <organization_name>. Écrit comme une *chaîne de caractères externes*. L'attribut suivant peut être spécifié.

[Attribut]

reading: Prononciation du nom de l'entreprise, écrit comme un attribut *"Reading"*. Peut être omis.

Exemple:

```
<organization_name >ABCD Corporation</organization name>
```

9.22 "Address"

Format de données permettant de définir une adresse, un numéro de téléphone, une adresse électronique et d'autres informations. Toutes ces informations sont placées dans les éléments-enfants d'une balise <address_info> principale.

Éléments-enfants]

<postal_code> Permet de stocker le code postal sous la forme d'une *chaîne de caractères standard*. Peut être omis.

<address> Permet de stocker l'adresse sous la forme d'une *chaîne de caractères externes*. Peut être omis.

<telephone> Permet de stocker le numéro de téléphone sous la forme d'une *chaîne de caractères standard*. Peut être omis.

<fax> Permet de stocker le numéro de télécopie sous la forme d'une *chaîne de caractères standard*. Peut être omis.

- <mail_address> Permet de stocker l'adresse électronique sous la forme d'une *chaîne de caractères standard*. Peut être omis.
- <website> Permet de stocker l'URI de la page d'accueil sous la forme d'une *chaîne de caractères standard*. Peut être omis.
- <address_other_info> Permet de stocker des informations supplémentaires que les éléments précédents n'ont pas abordées, sous la forme d'une *chaîne de caractères externes*. Peut être omis. Si ces informations doivent être affichées, il convient d'afficher les espaces (0x0020) et les sauts de ligne (0x000D, 0x000A, 0x0D0A) en l'état, mais les tabulations (0x0009) comme des espaces.

Exemple:

```
<address_info>
  <postal_code>75008</postal_code>
  <address>xxx avenue des champs elysees, Paris, France </address>
  <telephone>01 53 xx xx xx</telephone>
  <mail_address>xxx@XXXXX.fr</mail_address>
  <website>http://www.XXXXX.fr</website>
</address_info>
```

9.23 "Permission"

Définit les autorisations, par exemple d'imprimer ou de copier. Les différentes autorisations sont stockées dans les éléments-enfants de la balise <permission_info>. Si la balise <permission_info> est omise, les autorisations prennent la valeur qu'elle auraient si la balise d'autorisation leur correspondant était omise. Les explications ci-dessous font référence à un "utilisateur authentifié" (*authenticated user*). Dans des circonstances habituelles, tous les lecteurs du document sont supposés authentifiés. Toutefois, le format de distribution peut contenir des technologies DRM et des mécanismes d'authentification. Les éléments-enfants ci-dessous sont disponibles.

[Éléments-enfants]

<print_permission> Permet d'indiquer si l'impression est autorisée ou interdite. S'il est omis, la valeur par défaut de l'attribut permission est "no".

[Attribut]

permission: Permet d'indiquer si l'impression est autorisée ou interdite. Les valeurs suivantes sont possibles:

"authorized": les utilisateurs authentifiés peuvent imprimer.

"no": personne ne peut imprimer (valeur par défaut).

<copy_permission> Permet d'indiquer si la copie est autorisée ou interdite. S'il est omis, la valeur par défaut de l'attribut permission est "no".

[Attribut]

permission: Permet d'indiquer si la copie est autorisée ou interdite. Les valeurs suivantes sont possibles:

"authorized": les utilisateurs authentifiés peuvent copier.

"in_device_only": les utilisateurs authentifiés peuvent copier, mais uniquement dans le dispositif de visualisation. Si le dispositif n'offre pas de mécanismes

"no":

permettant d'empêcher les copies externes, la copie est interdite.

personne ne peut copier (valeur par défaut).

Exemple:

```
<permission_info>
  <print_permission permission="authorized"/>
</permission_info>
```

9.24 "Keyword"

Avec la balise `<keyword_list>`, il est possible d'associer une liste de mots clés aux données bibliographiques, à un flot de données ou à un objet (dans la présente spécification, elle est limitée aux données bibliographiques). Il convient qu'une balise `<keyword_list>` contienne une ou plusieurs balises `<keyword>` comme élément-enfant, chacune d'elles enregistrant un mot clé. La balise `<keyword>` comporte les attributs et l'élément-enfant suivants.

[Attributs]

category: Permet de définir la catégorie du mot clé, écrite sous la forme d'une *chaîne de caractères standard*. Peut être omis.

reading: Permet d'enregistrer la prononciation du mot clé, écrite sous la forme d'un attribut *Reading*. Peut être omis.

[Éléments-enfants]

Chaîne de caractères externes: Permet d'enregistrer le mot clé actuel.

Exemple:

```
<keyword_list>
  <keyword category="History" >Renaissance</keyword>
  <keyword>xml</keyword>
</keyword_list>
```

9.25 "Telephone_number"

Format de données permettant d'enregistrer des numéros de téléphone. Il permet de composer un numéro lorsque le dispositif de visualisation est un téléphone. Il est écrit sous la forme d'une combinaison de caractères figurant dans le Tableau 3. Le nombre de caractères doit être compris entre 1 et 64 inclus.

Si une fonction définie par le numéro de téléphone ne peut pas être exécutée, il convient que le dispositif de visualisation ne compose pas le numéro. À savoir:

lorsque le numéro de téléphone est trop long pour que le dispositif puisse l'employer,

lorsque le numéro de téléphone contient des caractères ne figurant pas dans le Tableau 3,

lorsque le numéro de téléphone contient une fonction qui ne peut pas être exécutée.

Tableau 3 – Caractères utilisables pour un numéro de téléphone

Caractère	ASCII code	Signification
0 à 9	0x30 à 0x39	numéro
#	0x23	bouton #
*	0x2A	bouton *
-	0x2D	Omis.
,	0x2C	Pause (1 seconde). Si on ne peut pas réaliser une pause de 1 seconde, attendre la pression sur la touche à la place.
/	0x2F	Pause (attente de pression sur la touche).
P ou p	0x50 ou 0x70	Pause (attente du signal). S'il n'est pas possible d'attendre le signal, attendre une pression sur la touche à la place.
+	0x2B	Signal d'appel international. Uniquement au début d'un numéro de téléphone (s'il est entré à un autre endroit, ne pas lancer l'appel.)

9.26 "Mail_address"

Format de données permettant de stocker une adresse électronique. Écrit sous la forme local-part@domain. local-part et domain peuvent utiliser les caractères du Tableau 4. La longueur maximale de local-part@domain est de 256 octets. Dans les caractères ci-dessous, "&" (0x26) doit être écrit comme une entité XML: "&". Ces entités XML comptent comme un octet.

Tableau 4 – Caractères utilisables pour des adresses électroniques

Catégorie	Caractères	ASCII code
chiffres	0 à 9	0x30 à 0x39
alphabet	A à Z	0x41 à 0x5A
	a à z	0x61 à 0x7A
!	!	0x21
\$	\$	0x24
%	%	0x25
&	&	0x26
*	*	0x2A
+	+	0x2B
-	-	0x2D
.	.	0x2E
/	/	0x2F
=	=	0x3D
?	?	0x3F
^	^	0x5E
_	_	0x5F
~	~	0x7E

10 Détails du format de description

10.1 Généralités

Chaque livre se présente sous la forme d'un document XML (voir ci-dessous).

```
<?xml version="1.0" encoding="UTF-8" ?>

<bvf id="1234" id-type="...">

    <book_info>
    ...           <!-- La bibliographie se trouve ici -->
    </book_info>

    <body_module>
    ...           <!-- Les données de flot se trouvent ici -->
    </body_module>

    <parts_module>
    ...           <!-- Les objets sont enregistrés ici -->
    </parts_module>

</bvf>
```

Le codage de caractères utilisé dans le document est spécifié en langage XML habituel, avec l'attribut de codage de la balise <xml>. Afin d'éviter les problèmes de conversion, la présente norme recommande l'utilisation d'UTF-8 ou d'UTF-16 (voir A.2.8 pour plus de détails).

10.2 Modules d'information du livre <bvf>

10.2.1 Généralités

Le module d'information du livre est enregistré dans la balise <bvf>. Il fait office de balise de racine pour toutes les données du livre, toutes les informations relatives au document y étant stockées (il peut également arriver que seuls les noms de fichiers externes soient stockés ici).

La balise <bvf> comporte les attributs et éléments-enfants suivants.

[Attributs]

- id_type: Permet de définir le type de numéro stocké dans l'attribut id. Écrit comme une *chaîne de caractères standard*. Peut être omis.
- id: Permet d'enregistrer le numéro d'identification de ce livre dans le système spécifié par l'attribut id_type. Écrit comme une *chaîne de caractères standard*. Peut être omis. Ne pas confondre avec Reference ID (voir 10.6.3.4).
- default_ccs: Permet de définir le nom du jeu de caractères des *caractères standard* et des *caractères étendus* utilisés dans cette norme (voir Article 9). Si l'on spécifie plus d'un jeu de caractères, on les sépare d'une "," (0x002C). Cet attribut ne doit pas être omis.
- display_size: Permet de préciser la taille de l'écran supposée lors de la création du contenu. Écrite au format de *coordonnées standard*. Peut être omis.

[Éléments-enfants]

- <book_info> Permet d'enregistrer les données bibliographiques. Cet attribut ne doit pas être omis. Voir 10.2.2 pour plus de détails.

<body_module> Module de gestion du contenu. Cet attribut ne doit pas être omis. Voir 10.3 pour plus de détails.

<parts_module> Modules des données de pièces. Cet attribut ne doit pas être omis. Voir 10.5 pour plus de détails.

Exemple:

```
<bvf id_type="ISBN" id="xxx-x-xxxx-xxxx-x" default_ccs="ISO 646-IRV">
    <book_info> ... </book_info>          <!-- Bibliographie -->
    <body_module> ... </body_module>      <!-- Module de contenu -->
    <parts_module> ... </parts_module>    <!-- Modules des données des parties -->
</bvf>
```

10.2.2 Données bibliographiques <book_info>

Permet de stocker les données bibliographiques (l'auteur ou le titre, par exemple). Cette balise <book_info> comporte les éléments-enfants suivants.

[Éléments- enfants]

<title_info> Permet de stocker les informations relatives au titre. Cet attribut ne doit pas être omis. Ses éléments-enfants sont les suivants.

[Éléments-enfants]

<series_title> Le titre de la série est enregistré sous la forme d'une *chaîne de caractères externes* dans cet élément. En l'absence de titre de série, il peut être omis. L'attribut suivant peut être utilisé.

[Attribut]

reading: permet de donner la prononciation du titre de la série sous la forme d'un attribut *Reading*. Peut être omis.

<title> Le titre est enregistré sous la forme d'une *chaîne de caractères externes* dans cet élément. Il ne doit pas être omis. L'attribut suivant peut être utilisé.

[Attribut]

reading: permet de donner la prononciation du titre sous la forme d'un attribut *Reading*. Peut être omis.

<subtitle> Le sous-titre est enregistré sous la forme d'une *chaîne de caractères externes* dans cet élément. En l'absence de sous-titre, il peut être omis. L'attribut suivant peut être utilisé.

[Attribut]

reading: permet de donner la prononciation du sous-titre sous la forme d'un attribut *Reading*. Peut être omis.

<edition_info> Les informations relatives à l'historique de révision du livre sont enregistrées sous la forme d'une *chaîne de caractères externes* dans cet élément. Peut être omis. Si les informations doivent être affichées, le caractère d'espace (0x0020) et les caractères de saut de ligne et de

retour chariot (0x000D, 0x000A, 0x0D0A) doivent être affichés en l'état, mais le caractère de tabulation (0x0009) doit l'être comme s'il s'agissait d'un espace. L'attribut suivant peut être utilisé.

[Attribut]

this_version: Permet de préciser laquelle des versions citées est la version en vigueur. Par exemple: "Troisième révision". Peut être omis.

<title_other_info> D'autres informations relatives au titre peuvent être stockées sous la forme d'une *chaîne de caractères externes* dans cet élément. Peut être omis. Si les informations doivent être affichées, le caractère d'espace (0x0020) et les caractères de saut de ligne et de retour chariot (0x000D, 0x000A, 0x0D0A) doivent être affichés en l'état, mais le caractère de tabulation (0x0009) doit l'être comme s'il s'agissait d'un espace.

<author_info> Permet de stocker les informations relatives à l'auteur. Peut être omis. Chaque auteur est enregistré dans un élément-enfant **<author>** distinct.

La balise **<author>** comporte l'attribut et les éléments-enfants suivants.

[Attribut]

role: permet de spécifier le rôle de la personne mentionnée. Les valeurs possibles sont indiquées ci-dessous. S'il est omis, sa valeur par défaut est "author".

"author", "editor", "translator", "supervisor", "designer",
"photographer", "illustrator", "binder", "planner", "other"

[Éléments-enfants]

<personal_name> / **<organization_name>**

Permet d'enregistrer le nom de l'auteur dans l'une de ces deux balises, selon qu'il s'agit d'une personne ou d'une organisation. Respectivement écrits en tant que *Personal_name* ou *Organization_name*.

<address_info> Adresse de l'auteur, écrite au format de données *Address*. Peut être omis.

<author_other_info> D'autres informations relatives à l'auteur peuvent être stockées sous la forme d'une *chaîne de caractères externes* dans cet élément. Peut être omis. Si les informations doivent être affichées, le caractère d'espace (0x0020) et les caractères de saut de ligne et de retour chariot (0x000D, 0x000A, 0x0D0D) doivent être affichés en l'état, mais le caractère de tabulation (0x0009) doit l'être comme s'il s'agissait d'un espace.

<publisher_info> Permet de stocker les informations relatives à l'éditeur. Peut être omis. La balise **<publisher_info>** comporte les éléments-enfants suivants. Si **<publisher_info>** est spécifié, l'un au moins de **<publisher>** ou **<publisher_office>** doit l'être également.

[Éléments-enfants]

<publisher> Permet de stocker des informations relatives à l'éditeur, s'il s'agit d'une personne. Peut être omis. Ses éléments-enfants sont les suivants.

[Éléments-enfants]

<publisher_name> Le nom de l'éditeur est enregistré sous la forme d'une *chaîne de caractères externes* dans cet élément. Cet attribut ne doit pas être omis. L'attribut suivant peut être utilisé.

[Attribut]

reading: permet de donner la prononciation du nom de l'éditeur sous la forme d'un attribut *Reading*. Peut être omis.

<address_info> Permet de stocker l'adresse de l'éditeur, au format *Address* standard. Peut être omis.

<publisher_office> Permet de stocker des informations relatives à l'éditeur, s'il s'agit d'une entreprise. Peut être omis. Ses éléments-enfants et attributs sont les suivants.

[Attribut]

publisher_code: Permet d'enregistrer l'ID de l'éditeur. Peut être omis.

[Éléments-enfants]

<organization_name> Le nom de l'organisation est enregistré dans l'élément. Peut être omis.

<address_info> L'adresse de l'éditeur est enregistrée dans cet élément, au format *Address* standard. Peut être omis.

<publisher_other_info> D'autres informations relatives à l'éditeur peuvent être stockées sous la forme d'une *chaîne de caractères externes* dans cet élément. Peut être omis. Si les informations doivent être affichées, le caractère d'espace (0x0020) et les caractères de saut de ligne et de retour chariot (0x000D, 0x000A, 0x0D0A) doivent être affichés en l'état, mais le caractère de tabulation (0x0009) doit l'être comme s'il s'agissait d'un espace.

<seller_info> Permet de stocker les informations relatives au vendeur. Peut être omis. La balise <seller_info> comporte les éléments-enfants suivants. Si <seller_info> est spécifié, l'un au moins de <seller> ou <seller_office> doit l'être également.

[Éléments-enfants]

<seller> Permet de stocker des informations relatives au vendeur, s'il s'agit d'une personne. Peut être omis. Ses éléments-enfants sont les suivants.

[Éléments-enfants]

<seller_name> Le nom du vendeur est enregistré sous la forme d'une *chaîne de caractères externes* dans cet élément. Cet attribut

ne doit pas être omis. L'attribut suivant peut être utilisé.

[Attribut]

reading: permet de donner la prononciation du nom du vendeur sous la forme d'un attribut *Reading*. Peut être omis.

<address_info> Permet de stocker l'adresse du vendeur, au format *Address* standard. Peut être omis.

<seller_office> Permet de stocker des informations relatives au vendeur, s'il s'agit d'une entreprise. Peut être omis. Ses éléments-enfants et attributs sont les suivants.

[Attribut]

seller_code: Permet d'enregistrer l'ID du vendeur. Peut être omis.

[Éléments-enfants]

<organization_name> Permet d'enregistrer le nom de l'organisation. Peut être omis.

<address_info> Permet d'enregistrer l'adresse du vendeur, écrite au format *Address* standard. Peut être omis.

<seller_other_info> D'autres informations relatives au vendeur peuvent être stockées sous la forme d'une *chaîne de caractères externes* dans cet élément. Peut être omis. Si les informations doivent être affichées, le caractère d'espace (0x0020) et les caractères de saut de ligne et de retour chariot (0x000D, 0x000A, 0x0D0A) doivent être affichés en l'état, mais le caractère de tabulation (0x0009) doit l'être comme s'il s'agissait d'un espace.

<book_id_info> Permet d'enregistrer le numéro d'identification des livres (le numéro ISBN, par exemple). Peut être omis. Son élément-enfant est le suivant.

[Éléments-enfants]

<book_id> Chaque type de numéro d'identification est stocké dans une balise <book_id> sous la forme d'une *chaîne de caractères standard*. Si <book_id_info> n'est pas omis, il doit exister au moins un <book_id>. Une valeur doit être affectée à l'attribut suivant.

[Attribut]

type: Permet de spécifier le type de numéro d'identification ("ISBN", par exemple). Écrit comme une *chaîne de caractères standard*. Cet attribut ne doit pas être omis.

<classification_info> Permet de stocker des informations relatives à la classification du livre. Peut être omis. Son élément-enfant est le suivant.

[Éléments-enfants]

<classification> Chaque type différent de classification est stocké dans une balise <classification> distincte. Si <classification_info> n'est pas omis, il doit exister au moins une <classification>. Il est stocké sous la forme d'une *chaîne de caractères externes*. Une valeur doit être affectée à l'attribut suivant.

[Attribut]

type: type de classification utilisé. Cet attribut ne doit pas être omis.

<rating> Permet de classer le contenu comme violent ou destiné aux adultes. Peut être omis. Les attributs suivants peuvent être utilisés.

[Attributs]

adult: Permet d'indiquer que le contenu est destiné aux adultes. Les valeurs possibles sont "yes" ou "no". La valeur par défaut est "no" en cas d'omission.

violence: Permet de classer le contenu comme violent. Les valeurs possibles sont "yes" ou "no". La valeur par défaut est "no" en cas d'omission.

<publication_place> Le pays de publication est enregistré comme un *Country* standard dans cet élément. Peut être omis.

<publication_date_info> Permet de stocker des informations relatives à la date de publication du livre. Peut être omis. Chaque date doit être stockée dans une instance distincte de l'élément-enfant suivant.

[Éléments-enfants]

<publication_date> Permet de stocker une date pertinente de publication (la date de publication elle-même ainsi que d'autres dates comme la date d'impression ou de début des ventes, etc.). Chaque date est stockée au format *Date* standard. Si <publication_date_info> n'est pas omis, il doit exister au moins un <publication_date>. L'attribut suivant peut être utilisé.

[Attribut]

type: Permet de spécifier le type de date dont il s'agit. "publish" si c'est la date de publication et "sale" si c'est la date à laquelle la vente commence. S'il est omis, sa valeur par défaut est "publish".

<net_price_info> Permet de définir le prix du livre. Peut être omis, si le prix est libre ou non défini. Plusieurs prix peuvent être stockés dans des instances distinctes de l'élément-enfant suivant, par devise et par pays.

[Éléments-enfants]

<net_price> Permet de stocker un prix spécifique à un pays et une devise, écrit sous la forme d'une *chaîne de caractères standard*. Si <net_price_info> n'est pas omis, il doit exister au moins un <net_price>. Les attributs suivants peuvent être utilisés.

[Attributs]

country: Permet de définir le pays dans lequel il convient d'appliquer ce prix, au format

Country standard. S'il est omis, le prix s'applique à tous les pays.

unit: Permet de définir la devise sous la forme d'une *chaîne de caractères standard*. Cet attribut ne doit pas être omis.

other_info: autres informations, écrites comme une *chaîne de caractères standard*. Peut être omis.

<book_abstract> Un résumé du livre, écrit sous la forme d'une *chaîne de caractères externes*, est enregistré dans cet élément. Peut être omis. Si les informations doivent être affichées, le caractère d'espace (0x0020) et les caractères de saut de ligne et de retour chariot (0x000D, 0x000A, 0x0D0A) doivent être affichés en l'état, le caractère de tabulation (0x0009) devant l'être comme s'il s'agissait d'un espace.

<front_cover_image> Permet de définir l'image à utiliser comme page de couverture, en enregistrant son emplacement sous un *Nom de fichier* standard. Actuellement, les images jpeg et png sont prises en charge. Peut être omis. Le type de fichier tel que défini par cet attribut doit être comparé à l'attribut de type avant d'ouvrir le fichier. L'attribut suivant doit être employé.

[Attribut]

type: Permet de définir le type de l'image, en donnant son type MIME ("image/jpeg", par exemple). Cet attribut ne doit pas être omis.

<spine_cover_image> Permet de définir l'image à utiliser comme image spine. Suit les mêmes règles que <front_cover_image>. Peut être omis.

<keyword_list> Permet d'enregistrer une liste des mots clés relatifs aux données du livre, écrite dans la syntaxe *Keyword*. Peut être omis.

<other_book_info> D'autres informations relatives au livre peuvent être stockées sous la forme d'une *chaîne de caractères externes* dans cet élément. Peut être omis. Si les informations doivent être affichées, le caractère d'espace (0x0020) et les caractères de saut de ligne et de retour chariot (0x000D, 0x000A, 0x0D0A) doivent être affichés en l'état, le caractère de tabulation (0x0009) devant l'être comme s'il s'agissait d'un espace.

Example:

```
<book_info>
  <title_info>
    <series_title>Dummy books</series_title>
    <title>the dummy book of nonsense</title>
    <edition_info>2000/01/01 first edition,
                  2005/01/01 second edition</edition_info>
  </title_info>
  <author_info>
    <author role="author">
      <personal_name>
        <first_name>John</first_name>
        <last_name>Smith</last_name>
      </personal_name>
      <address_info>
        <mail_address>john.smith@abcd.com</mail_address>
        <website>http://www.abcd.com/~jsmith/</website>
      </address_info>
    </author>
  </author_info>
  <publisher_info>
    <publisher_office>
      <organization_name>abcd corporation</organization_name>
      <address_info>
        <postal_code>100-1000</postal_code>
        <address>1 main street, Fooobar city, Japan</address>
      </address_info>
    </publisher_office>
  </publisher_info>
  <book_id_info>
    <book_id type="ISBN">xxx-x-xxxx-xxxx-x</book_id>
    <book_id type="Japanese_ID_number">454745-7</book_id>
  </book_id_info>
  <classification_info>
    <classification type="Japanese_C_CODE">2143</classification>
  </classification_info>
  <rating adult="no" violence="no"/>
  <publication_place>jpn</publication_place>
  <publication_date_info>
    <publication_date type="publish">2005</publication_date>
  </publication_date_info>
  <net_price_info>
    <net_price country="jpn" unit="yen">1200</net_price>
  </net_price_info>
  <book_abstract>This book doesn't talk about anything special.</book_abstract>
  <front_cover_image type="image/png">xxx.png</front_cover_image>
  <spine_cover_image type="image/png">yyy.png</spine_cover_image>
  <keyword_list>
    <keyword>dummy</keyword>
    <keyword>nonesense</keyword>
  </keyword_list>
</book_info>
```

10.3 Module de gestion de contenu <body_module>

10.3.1 Généralités

Le module de gestion de contenu (<body_module>) est en charge de la coordination du contenu dans la fabrication du document réel. Son élément-enfant est le suivant.

[Attribut]

Aucun

[Éléments-enfants]

<flow_type_body> Permet de traiter les données du contenu courant. Voir 10.3.2 pour plus de détails.

Exemple:

```
<body_module>
  <flow_type_body>
    <flow_entry>
      <flow_data ...> ... </flow_data>
      <flow_data ...> ... </flow_data>
      ...
    </flow_entry>
    <special_page_link>
      <special_page kind="contents">PG0001</special_page>
      <special_page kind="body">PG0002</special_page>
    </special_page_link>
    <search_table>
      ...
    </search_table>
  </flow_type_body>
</body_module>
```

10.3.2 Données de contenu courant <flow_type_body>

10.3.2.1 Balise <flow_type_body>

La balise <flow_type_body> traite les données du contenu courant. Ses éléments-enfants sont les suivants.

[Attribut]

Aucun

[Éléments-enfants]

<flow_entry> Permet d'enregistrer les données de flot à utiliser comme contenu courant du texte principal. Il ne doit exister qu'une seule instance de cet élément. Cet attribut ne doit pas être omis. Voir 10.3.2.2 pour plus de détails.

<special_page_link> Données de page spéciale. Permet de spécifier la position du contenu courant de certaines pages indispensables (l'index ou le début du contenu principal, par exemple), afin de faciliter la référence. Peut être omis. Écrit comme indiqué en 10.3.2.5. L'omission de cet élément signifie qu'il n'existe aucune page spéciale.

<search_table> Permet d'enregistrer les données nécessaires à la création d'une table de recherche. Peut être omis. Écrit comme indiqué en 10.3.2.6. L'omission de cet élément signifie qu'il n'existe aucune table de recherche.

10.3.2.2 Module d'enregistrement des données de flot <flow_entry>

La balise <flow_entry> permet d'enregistrer les données de flot à utiliser comme contenu courant du texte principal. Ses Éléments-enfants sont les suivants.

[Éléments-enfants]

- <flow_default_attribute> Permet de définir les attributs par défaut à utiliser pour afficher chaque donnée de flot, telle que définie par l'élément <flow_data> suivant. Peut être omis. Voir 10.3.2.3 pour plus de détails. S'il est omis, il convient que le dispositif de visualisation se comporte comme si une valeur par défaut était affectée à l'ensemble de ses attributs et éléments-enfants.
- <flow_data> Permet d'enregistrer les informations relatives à chaque donnée de flot. Il existe une relation univoque entre le nombre de balises <flow_data> et les données de flot réelles à enregistrer/afficher. Il doit exister une ou plusieurs instances de cette balise. L'ordre d'enregistrement des données de flot détermine l'ordre d'affichage. Voir 10.3.2.4 pour plus de détails.

10.3.2.3 Module de l'attribut par défaut des données de flot <flow_default_attribute>

Permet de définir les valeurs par défaut des attributs à utiliser pour afficher les données de flot (voir 10.3.2.4) dont est composé le contenu courant principal. Les valeurs définies dans cette balise font office de valeurs par défaut pour tous les flots comportant un objet texte, un objet d'écran de recherche ou un objet données de dictionnaire comme données principales.

Une partie des valeurs qui peuvent être définies dans cette balise peut également être définie en local dans l'*instanciation d'objet texte* de chaque donnée de contenu. Pour définir la valeur par défaut de données de contenu particulières, il convient d'utiliser l'attribut <text_default_attribute> des *instanciations d'objet texte* (voir 10.6.2) ou <dict_default_attribute> de l'objet données de dictionnaire désigné par les données de contenu. Si la valeur définie par la balise <flow_default_attribute> de l'ensemble des données de contenu entre en conflit avec la valeur définie dans un attribut <text_default_attribute> ou <dict_default_attribute> de données de contenu individuel, cette dernière prévaut. Si aucune valeur n'est attribuée, le comportement n'est pas défini par la présente norme et dépend de la valeur par défaut du dispositif de visualisation ou des préférences de l'utilisateur. De plus, si le dispositif de visualisation ne met pas en œuvre la méthode requise ou s'il souhaite donner la priorité aux paramètres choisis par l'utilisateur, il peut le faire sans tenir compte de la valeur définie dans ces balises de valeur par défaut, sauf si les explications ci-après indiquent le contraire.

La balise <flow_default_attribute> comporte les attributs et éléments-enfants suivants. Si elle est omise, il convient que le dispositif de visualisation se comporte comme si leur propre valeur par défaut était affectée à l'ensemble de ses attributs et éléments-enfants.

[Attributs]

- baseline: Permet de définir l'orientation de la ligne de base (et donc du texte) de chaque flot. Les valeurs possibles sont indiquées ci-dessous. Peut être omis. Si aucune valeur n'est affectée à cet attribut ou à l'option correspondante dans chaque donnée de contenu (l'attribut baseline de la balise <text_default_attribute> de l'*instance d'objet texte* ou <dict_default_attribute> de l'instance d'objet données de dictionnaire, voir 10.6.2.2 et 10.6.3.2 pour plus de détails), la valeur par défaut dépend du dispositif de visualisation.
 - "right": Le sens d'écriture est horizontal (de gauche à droite). Toutefois, il peut être modifié par l'utilisateur.
 - "right_only": Le sens d'écriture est horizontal (de gauche à droite), et l'utilisateur ne peut pas modifier le paramètre. Toutefois, s'il n'est pas pris en charge par le dispositif de

	visualisation, ce paramètre n'est pas nécessairement appliqué.
"down":	Le sens d'écriture est vertical (de haut en bas). Toutefois, il peut être modifié par l'utilisateur.
"down_only":	Le sens d'écriture est vertical (de haut en bas), et l'utilisateur ne peut pas modifier le paramètre. Toutefois, s'il n'est pas pris en charge par le dispositif de visualisation, ce paramètre n'est pas nécessairement appliqué.
view_type:	Permet de définir l'orientation par défaut de l'écran pour chaque flot. Les valeurs possibles sont indiquées ci-dessous. Peut être omise. Si cette valeur est omise, la valeur par défaut dépend du dispositif de visualisation.
"portrait":	permet de choisir du mode portrait (plus haut que large). Toutefois, il peut être aligné sur une autre direction au choix de l'utilisateur.
"portrait_only":	permet de choisir le mode portrait (plus haut que large), l'utilisateur ne pouvant pas modifier le paramètre. Toutefois, si le dispositif de visualisation ne peut pas traiter cette orientation d'écran, ce paramètre n'est pas nécessairement appliqué.
"landscape":	permet de choisir le mode paysage (plus large que haut). Toutefois, il peut être aligné sur une autre direction au choix de l'utilisateur.
"landscape_only":	permet de choisir le mode paysage (plus large que haut), l'utilisateur ne pouvant pas modifier le paramètre. Toutefois, si le dispositif de visualisation ne peut pas traiter cette orientation d'écran, ce paramètre n'est pas nécessairement appliqué.
lang:	Permet de définir la langue par défaut pour chaque flot.
lang_code:	Permet de définir le code de langue par défaut du contenu. La valeur par défaut est "ISO 639-3". Il s'agit de préciser une langue absente des systèmes de code de langue existants. Dans la mesure du possible, il est recommandé de se conformer à l'ISO 639-3. Il peut être remplacé par l'attribut lang_code de la balise .
	"ISO 639-3": L'attribut de langue est écrit conformément à l'ISO 639-3.
	Autres valeurs: Représente le système de code de langue au choix de l'utilisateur.

Exemple:

```
<flow_default_attribute base_line="right" view_type="portrait"
lang_code = "ISO 639-3" , lang = "jpn">
```

phonetic_notation: Permet de définir la notation de prononciation par défaut du contenu. La valeur par défaut est "IPA". Elle est remplacée par l'attribut de notation phonétique des balises <pronunciation> et <headword> à l'intérieur de ce type de balise.

[Éléments-enfants]

<flow_default_size> Permet de définir l'interlettrage, l'espacement entre lignes et la dimension de la marge de tous les flots des données de contenu. Peut être omis. Si c'est le cas, il convient que le dispositif de visualisation se comporte comme si une valeur par défaut était affectée à tous ces attributs. Les attributs de cette balise sont les suivants.

[Attributs]

letter_spacing: Interlettrage par défaut. Les valeurs suivantes sont admises. Peut être omis. Si c'est le cas, la dimension n'est pas précisée et dépend du dispositif de visualisation.

"maximum"

"big"

"medium"

"small"

"minimum"

Les dimensions réelles de ces 5 possibilités dépendent du dispositif de visualisation et des capacités du dispositif sous-jacent.

line_pitch: Espacement entre lignes par défaut. Les valeurs suivantes sont admises. Peut être omis. Si c'est le cas, la dimension n'est pas précisée et dépend du dispositif de visualisation.

"maximum"

"big"

"medium"

"small"

"minimum"

Les dimensions réelles de ces 5 possibilités dépendent du dispositif de visualisation et des capacités du dispositif sous-jacent.

margin: Dimension de la marge par défaut. Les valeurs suivantes sont admises. Peut être omis. Si c'est le cas, la dimension n'est pas précisée et dépend du dispositif de visualisation.

"big"

"medium"

"small"

Les dimensions réelles de ces 3 possibilités dépendent du dispositif de visualisation et des capacités du dispositif sous-jacent.

<flow_default_font> Permet de définir le nom de la police, la taille et les propriétés de tous les flots des données de contenu. Peut être omis. Si un flot individuel de données de contenu le définit aussi (avec la balise <text_default_font> de l'instanciation d'objet texte, comme indiqué en 10.6.2.2 et la balise <dict_default_font> de l'instanciation d'objet données de dictionnaire comme indiqué en 10.6.3.2, les valeurs individuelles prévalent. S'ils sont tous deux omis, il convient que le comportement corresponde à la valeur par défaut indiquée ci-dessous. Les attributs suivants peuvent être utilisés.

[Attributs]

fontname: Nom de police par défaut. Plusieurs polices peuvent être spécifiées. Dans ce cas, il convient de les séparer par une virgule (0x2c en Unicode). Par exemple:

fontname="Aaa sans serif,Bbb gothic"

Il convient que le dispositif de visualisation utilise la première police répertoriée disponible. Peut être omis. Si cet attribut et l'attribut du flot individuel sont omis, la valeur par défaut dépend du dispositif de visualisation.

fontsize:	Taille de police par défaut. Les valeurs suivantes sont admises. Peut être omis. Si c'est le cas, la dimension n'est pas précisée et dépend du dispositif de visualisation. "maximum" "big" "medium" "small" "minimum" Les dimensions réelles de ces 5 possibilités dépendent du dispositif de visualisation et des capacités du dispositif sous-jacent.
bold_flag:	Permet de spécifier s'il convient d'afficher les données de contenu en gras ou non. Si cet attribut est omis, le comportement dépend du dispositif de visualisation. Les valeurs admises sont: "yes": afficher en gras "no": afficher normalement Si la valeur "yes" est attribuée, tous les caractères doivent s'afficher en gras, sauf, dans une <i>instance d'objet texte</i> ou une instance d'objet données de dictionnaire, ceux de la balise dont la valeur "no" a été affectée à l'attribut bold_flag (voir 10.6.2.3 et 10.6.3.4).
color_space, color, opacity:	Permet de définir la couleur de la police à utiliser pour les données de contenu. Écrit au format de données <i>color</i> standard. Si cet attribut est omis, la valeur dépend du dispositif de visualisation.
ruby_flag:	Ce drapeau "ruby" permet de définir si le code en langage de programmation "ruby" dans les données de contenu doit être ou non visualisé. S'il est omis, le comportement dépend du dispositif de visualisation. Le texte à afficher lorsque "ruby" est actif est celui placé dans la balise <ruby> des instanciations d'objet texte ou des instanciations d'objet données de dictionnaire. Les attributs suivants peuvent être utilisés. "yes": Il convient d'afficher le code "ruby", mais il peut être désactivé par l'utilisateur. "yes_only": Le code "ruby" doit être affiché et ne peut pas être désactivé par l'utilisateur. Toutefois, cela ne concerne pas les dispositifs de visualisation qui ne sont pas en mesure d'afficher le code ruby. "no": Il convient de ne pas afficher le code "ruby", mais il peut toujours être activé par l'utilisateur. "no_only": Le code "ruby" ne doit pas être affiché et ne peut pas être activé par l'utilisateur. Toutefois, cela ne s'applique pas si le dispositif de visualisation est incapable
italic:	Permet de décider s'il convient d'afficher les données de contenu en italique ou non. Si cet attribut est omis, le comportement dépend du dispositif de visualisation. Les valeurs admises sont:

"yes": afficher en italique
 "no": afficher normalement

Si la valeur "yes" est attribuée, tous les caractères doivent s'afficher en italique, sauf, ceux qui dans une *instanciation d'objet texte* ou une instanciation d'objet données de dictionnaire, ont leur balise avec une valeur "no" affectée à l'attribut italic (voir 10.6.2.3 et 10.6.3.4).

oblique: Permet de décider s'il convient d'afficher les données de contenu en oblique ou non. Si cet attribut est omis, le comportement dépend du dispositif de visualisation. Les valeurs admises sont

"yes": afficher en oblique;
 "no": afficher normalement.

Si la valeur "yes" est attribuée, tous les caractères doivent s'afficher en oblique, sauf, ceux qui dans une *instanciation d'objet texte* ou une instanciation d'objet données de dictionnaire, ont leur balise avec une valeur "no" affectée à l'attribut oblique (voir 10.6.2.3 et 10.6.3.4).

small_caps: Permet de décider s'il convient d'afficher les données de contenu en petites capitales ou non. Si cet attribut est omis, le comportement dépend du dispositif de visualisation. Les valeurs admises sont

"yes": afficher en petites capitales;
 "no": afficher normalement.

Si la valeur "yes" est attribuée, tous les caractères doivent s'afficher en petites capitales, sauf, ceux qui dans une *instanciation d'objet texte* ou une instanciation d'objet données de dictionnaire, ont leur balise avec une valeur "no" affectée à l'attribut small_caps (voir 10.6.2.3 et 10.6.3.4).

family: Permet de spécifier la famille de polices à utiliser. La police utilisée pour le rendu dépend du dispositif de visualisation. Si cet attribut est omis, le comportement dépend également du dispositif de visualisation. Les valeurs admises sont

"monospace": afficher dans les polices de largeur fixe;
 "sans-serif": afficher dans les polices sans serifs;
 "serif" afficher dans les polices avec serifs;
 "cursive" afficher dans les polices en style manuscrit.

"fantasy" afficher dans les polices décoratives
 Si la valeur "yes" est attribuée, tous les caractères doivent s'afficher en style décoratif, sauf, ceux qui dans une *instanciation d'objet texte* ou une instanciation d'objet données de dictionnaire, ont leur balise avec une valeur "no" affectée à l'attribut fantasy (voir 10.6.2.3 et 10.6.3.4).

ul_type Permet d'indiquer la manière dont il convient de rendre les chaînes de la balise <u> et son équivalent (). Les valeurs acceptables sont les suivantes. Elles sont toutes issues de l'ISO/CEI 9541-1 (8.7.1.12.1.1 Score Name). Voir la présente norme pour connaître le sens exact de ces valeurs.

"rightscore"

	"leftscore"
	"throughscore"
em_type:	Permet de spécifier la manière dont il convient de rendre les chaînes des balises . Les valeurs acceptables se présentent comme suit, dont les quatre premières sont issues de l'ISO/CEI 9541-1: (8.7.1.12.1.1 Score Name). Voir le document pour connaître le sens exact de ces valeurs.
	Noter que lorsque cet attribut est spécifié, la balise n'est plus équivalente à .
	"rightscore"
	"leftscore"
	"throughscore"
	"kendot"
	"bold": Il convient de rendre les sous-chaînes en gras.
	"italic": Il convient de rendre les sous-chaînes en italique.
	"bold_italic" Il convient de rendre les sous-chaînes en gras italique
	"reverse" Il convient de rendre la sous-chaîne avec les caractères et l'arrière-plan inversés.
	"shade" Il convient de rendre les sous-chaînes avec une ombre.
<flow_default_background>	Permet de définir la couleur d'arrière-plan à utiliser pour tous les flots de données de contenu. Si un flot individuel de données de contenu le définit également (avec la balise <text_default_font> de l'instanciation d'objet texte, comme indiqué en 10.6.2.2 ou avec la balise <dict_default_font> de l'instanciation d'objet données de dictionnaire comme indiqué en 10.6.3.2), les valeurs individuelles prévalent. S'ils sont tous deux omis, il convient que le comportement corresponde à la valeur par défaut des attributs énumérés ci-dessous. Les attributs de cette balise sont les suivants.
[Attributs]	color_space, color, opacity:
	Permet de définir la couleur de police à utiliser pour les données de contenu. Écrit au format de données color standard. Si cet attribut est omis, la valeur dépend du dispositif de visualisation.
<flow_default_line_breaking_method>	Permet de spécifier l'algorithme à employer pour découper le texte en lignes. Les différentes langues traitent cela de diverses manières. Cet élément autorise une certaine souplesse. Peut être omis. Si cette balise et les balises <line_breaking_method> de l'instanciation d'objet texte/de données de dictionnaire (définies en 10.6.2.2 et 10.6.3.2) sont spécifiées, ces dernières prévalent. Si elles sont toutes omises, les valeurs par défaut de cette balise <flow_default_line_breaking_method> définissent le comportement. Cette balise comporte les attributs et éléments-enfants suivants.
[Attributs]	
method:	permet de choisir la méthode de saut de ligne. La valeur par défaut est "none". Les valeurs possibles sont:
	"none": aucun traitement particulier. Si une ligne est remplie par des caractères, passer à la suivante.

D'autres valeurs sont possibles pour la localisation. Voir Article A.2.

D'autres attributs peuvent être ajoutés par des méthodes "localisées" propres à une langue. Voir Article A.2.

[Éléments-enfants]

Aucun, à l'exception de ceux ajoutés par des méthodes "localisées" propres à une langue. Voir Article A.2.

<flow_default_delimiter> Permet d'indiquer le bloc des délimiteurs (chaînes de caractères placées avant le premier caractère et après le dernier caractère) des différents éléments du contenu. Ses éléments-enfants sont des balises qui spécifient ces délimiteurs pour certaines balises, qui servent de guide lorsque le contenu est converti en données pour le dispositif de visualisation ou rendu directement. **<flow_default_delimiter>** peut être omis, auquel cas la valeur par défaut de tous les délimiteurs est "" (chaîne vide).

[Éléments-enfants]

<delimiter> Permet de définir les délimiteurs de début et de fin de l'élément-enfant de la balise spécifiée par l'attribut de la balise.

Il est également possible de spécifier l'attribut de type, le cas échéant (c'est-à-dire que la balise spécifiée comporte l'attribut de type). La balise doit être omise si l'attribut de la balise spécifie une balise de rendu non textuel. Si aucun délimiteur n'est spécifié dans une balise, on considère qu'une chaîne vide "" est spécifiée.

[Attributs]

tag: Permet de spécifier la balise à laquelle s'appliquent les délimiteurs. Sera ignoré si une balise inexistante est spécifiée. Cet attribut ne doit pas être omis.

type: Permet de spécifier la valeur de l'attribut de type de la balise à laquelle s'appliquent les délimiteurs. Si l'attribut de type est spécifié dans une balise qui n'en comporte pas, l'instanciation de la balise **<delimiter>** est omise. Si plusieurs balises **<delimiter>** sont données, dont l'une comporte un attribut de type, tandis que les autres n'en n'ont pas, la première, si elle est applicable, remplace la dernière. Peut être omis.

open: Permet de définir le délimiteur de début (ouverture). Peut être omis et prendre la valeur par défaut "" (chaîne vide).

close: Permet de définir le délimiteur de fin (fermeture). Peut être omis et prendre la valeur par défaut "" (chaîne vide).

[Éléments-enfants]

Aucun.

Exemple:

```

< flow_default_delimiter>
<delimiter tag="headword" type="pronunciation" open="/" close="/"/>
<delimiter tag="etymology" open="&lt;" close="&gt;"/>
<delimiter tag="p" open="*" close="*"/>
</ flow_default_delimiter>

```

10.3.2.4 Données de flot <flow_data>

Les données de flot sont définies ici. L'objet à utiliser comme contenu du flot et d'autres informations (les liens de page et les événements, par exemple) sont enregistrés dans cette balise. Les attributs et éléments-enfants de la balise <flow_data> sont les suivants.

[Attributs]

- flow_id:** Permet de définir le numéro d'ID des données de flot, écrit sous la forme d'un *Page_ID*. Peut être omis. <flow_entry> ne doit contenir aucune autre donnée de flot portant le même ID.
- body_id:** Permet de pointer vers l'objet qui constitue le contenu de ce flot, utilisant l'*Object_ID* d'un objet "texte de contenu courant", d'un objet "page de recherche" ou d'un objet données de dictionnaire. Cet attribut ne doit pas être omis.
- turning_page_control:** Le dispositif de visualisation permet d'avancer ou de reculer dans le contenu. Toutefois, il est possible de limiter les déplacements vers le flot précédent ou suivant en affectant l'une des valeurs ci-dessous à cet attribut. S'il est omis, sa valeur par défaut est "off". Toutefois, noter que lorsqu'un ID de corps pointe vers un objet "page de recherche", la valeur "on" doit être attribuée à turning_page_control.
- "on":** Tout déplacement vers les flots précédents et suivants est interdit.
- "off":** Les déplacements vers les flots précédents et suivants sont autorisés.
- "forward":** Le déplacement vers le flot suivant est interdit, et vers le flot précédent autorisé.
- "back":** Le déplacement vers le flot suivant est autorisé, et vers le flot précédent interdit.

Comme expliqué ci-dessus, ce paramètre permet de limiter les déplacements d'un flot vers les autres mais pas les déplacements à l'intérieur d'un des flot. Pour limiter les déplacements à l'intérieur d'un objet "texte de contenu courant", la commande de changement de page de la balise <page_break> (voir 10.6.2.3) doit être utilisée dans l'objet texte de contenu courant et la balise <dic-item> de l'objet données de dictionnaire.

[Éléments-enfants]

- <event_info>** Module d'informations d'événement. Définit les événements (déclencheurs) et les actions associées. Peut être omis. Voir 10.4 pour plus de détails.

10.3.2.5 Données de page spéciale <special_page_link>

Permet d'enregistrer la position des pages importantes et fréquemment consultées, afin de faciliter l'accès à ces parties du document. Par exemple, il peut s'avérer utile d'accéder aisément à la carte d'un guide de la route, à la chronologie d'un livre d'Histoire ou au glossaire d'un document technique. Cette balise <special_page_link> comporte l'élément-enfant suivant.

[Éléments-enfants]

<special_page> Permet d'enregistrer les informations relatives à une seule page spéciale. Ses attributs et éléments-enfants sont les suivants.

[Attributs]

kind: Permet de décrire le contenu de la page en référence. Les valeurs possibles sont indiquées ci-dessous. S'il est omis, la valeur par défaut est "other".

"cover", "title_page", "preface", "contents" (table des matières), "body" (début du contenu), "column" (élément de texte encadré), "note", "figure", "ad", "afterword", "appendix", "answer", "glossary", "bibliography", "commentary", "index", "imprint", "author_info", "other".

title: Permet de définir un titre pour la position du document en référence comme une page spéciale. Écrit comme une chaîne de caractères étendus. Peut être omis.

[Éléments-enfants]

Chaîne de caractères standard: Permet de spécifier la position à considérer comme une page spéciale, en choisissant une donnée de flot particulière dans l'ensemble du contenu courant. Cet attribut ne doit pas être omis. Pour des données de flot textuelles, il est possible de spécifier une position à l'intérieur également.

- Pour spécifier uniquement les données de flot dans leur ensemble.

Écrit comme un *Page_ID*. Il permet d'enregistrer l'ID de page des données de flot correspondantes.

Exemple:

```
<special_page ... >PG0001</special_page>
```

- Pour spécifier les données de flot et la position à l'intérieur.

Écrit sous la forme "*Page_ID/Object_ID/Char_ID*". *Page_ID* est le numéro d'ID des données de flot. *Object_ID* est le numéro d'ID d'un objet texte/données de dictionnaire à l'intérieur de ce flot. *Char_ID* est le numéro d'ID d'une chaîne définie à l'intérieur de cet objet texte/données de dictionnaire.

Exemple:

```
<special_page ... >
    PG0001/OB0321/CR0982
</special_page>
```

Exemple:

```
<special_page_link>
  <special_page kind="contents">PG1111 </special_page>
  <special_page kind="other" title="Downtown area map">PG1114/OB0022/CR0001
  </special_page>
  ...
</special_page_link>
```

10.3.2.6 Données de la table de recherche <search_table>

Les informations définissant les tables de recherche y sont stockées, par exemple l'ID de la table de recherche ou les paramètres concernant les mots indexés enregistrés dans la table de recherche, "Mots indexés" signifient ci-dessous cible de chaque recherche; ils sont représentés de deux manières:

- enregistrés dans la balise <head> de l'*instance d'objet "texte"* (voir 10.6.2.3).
- enregistrés dans la balise <head> de l'*instanciation d'objet données de dictionnaire* (voir 10.6.3.4).

Lors de la conversion en format de distribution, et selon le paramètre stocké dans la balise <search_table_def> définie ci-dessous, la table de recherche réelle est construite, après vérification ou contrôle des chaînes de mots indexés. Dans le format de distribution, les mots indexés sont stockés dans cette table de recherche. La balise <search_table> comporte les attributs et éléments-enfants suivants.

[Attributs]

bookmark Permet de préciser si toutes les tables de recherche incluses dans le contenu sont admises à la mise automatique en signet sur le dispositif de visualisation (c'est-à-dire en conservant automatiquement le pointage sur les mots clés recherchés). Les valeurs suivantes sont possibles: Peut être omis et considéré comme "no" s'il est omis.

"yes": Les tables de recherche remplissent les conditions requises pour la mise automatique en signet.

"no": Les tables de recherche ne remplissent pas les conditions requises pour la mise automatique en signet.

wordbook Permet de préciser si toutes les tables de recherche sont admissibles à la fonction de livret sur le dispositif de visualisation (c'est-à-dire l'enregistrement automatique des mots clés dans le livret et l'affichage de la liste). Les valeurs suivantes sont possibles. Peut être omis et considéré comme "no" s'il est omis.

"yes": Les tables de recherche sont admissibles à la fonction de livret.

"no": Les tables de recherche ne sont pas admissibles à la fonction de livret.

jump_search_root Permet de préciser si le contenu est admissible comme source de recherche multi-contenu à partir du texte (c'est-à-dire si on peut lancer une recherche multi-contenu en utilisant le mot clé sélectionné dans le texte affiché du contenu). Les valeurs suivantes sont possibles. Peut être omis et considéré comme "no" s'il est omis.

"yes": le contenu est admissible comme source de recherche multi-contenu à partir du texte.

"no": le contenu n'est pas admissible comme source de recherche multi-contenu à partir du texte.

jump_search Permet de préciser si toutes les tables de recherche remplissent les conditions requises pour une recherche multi-contenu dans le texte (c'est-à-dire si on peut procéder à une recherche dans l'un des contenus de la recherche multi-contenu en utilisant le mot clé sélectionné dans le texte) et pour une recherche multi-contenu à partir d'une input lorsque le contenu est affiché (c'est-à-dire si on peut rechercher l'un des contenus dans une recherche multi-contenu en utilisant le mot clé employé comme input), avec le contenu affiché. Les valeurs suivantes sont possibles: Peut être omis et considéré comme "no" s'il est omis.

"yes": les tables de recherche remplissent les conditions requises pour ce type de recherches.

"no": les tables de recherche ne remplissent pas les conditions requises pour ce type de recherches.

all_search Permet de préciser si toutes les tables de recherche remplissent les conditions requises pour une recherche multi-contenu sans que le contenu lui-même soit affiché. Les valeurs suivantes sont possibles. Peut être omis et considéré comme "no" s'il est omis.

"yes": les tables de recherche remplissent les conditions requises pour ce type de recherches.

"no": les tables de recherche ne remplissent pas les conditions requises pour ce type de recherches.

Si cela est précisé dans les instanciations d'objet "texte", seule la valeur "no" est admise pour les attributs bookmark, wordbook, jump_search_root, jump_search et all_search.

[Éléments-enfants]

<search_table_def> Permet de spécifier les informations relatives à chaque table de recherche. La balise <search_table> doit comporter au moins une balise <search_table_def>. Les attributs et éléments-enfants ci-dessous sont disponibles.

[Attributs]

id: Permet de définir le numéro d'ID de cette table de recherche. Le numéro d'ID doit être unique à chaque table de recherche à l'intérieur du contenu et stocké sous la forme d'une *chaîne de caractères standard*. Cet attribut ne doit pas être omis. Lors de la conversion au format de distribution, la table de recherche est construite en fonction des paramètres de cette balise <search_table_def>, puis ajoutée au contenu. Ce numéro d'ID est utilisé pour faire référence à la table de recherche ou à partir des champs indexés de la page de recherche (voir 10.6.7 pour plus de détails) ou des informations de mot indexé stockées dans la balise <head> de l'instanciation d'objet texte (voir 10.6.2.3 pour plus de détails). Ne pas confondre avec "Reference ID" (voir 10.6.3.4).

use_default: Permet de définir si une recherche fondée sur cette table de recherche est admise par le dispositif de visualisation lorsque ladite recherche n'est pas lancée à partir d'une page de recherche connexe. Les attributs suivants peuvent être utilisés. S'il est omis, la valeur par défaut est "no".

"yes" cette recherche est admise.

"no" cette recherche n'est pas admise.

sorting_rule: Permet de définir la manière de trier les résultats de la recherche. Cet attribut consiste à proposer différents schémas de tri pour différentes langues. Peut être omis. La valeur par défaut est "unicode". Les attributs suivants peuvent être utilisés.

"unicode": Le rang de chaque caractère est déterminé par son point de code Unicode.

D'autres méthodes de tri sont définies dans la localisation linguistique (voir A.2.5).

name:	Permet de définir le nom de la table de recherche. Écrit dans une <i>chaîne de caractères standard</i> . Le nom est destiné à être utilisé dans la liste des résultats de la recherche. Cet attribut ne doit pas être omis.
short_name:	Permet de définir la version abrégée du nom de la table. Les noms plus longs que celui indiqué par l'attribut name ne sont pas admis. Le nom est destiné à être utilisé dans la liste des résultats de la recherche. Cet attribut ne doit pas être omis.
wild:	Permet de préciser s'il convient d'afficher la table de recherche pour la recherche approximative (c'est-à-dire que le nombre de lettres nécessaires pour remplir les vides du mot, ainsi que les positions des vides, sont connus). Les valeurs suivantes sont possibles. Cet attribut ne doit pas être omis. "yes" Il convient d'afficher la table de la recherche approximative. "no" sinon.
blank:	Permet de préciser s'il convient d'afficher la table de recherche pour la recherche de mot vide (où seules les positions des vides dans le mot sont connues). Les valeurs suivantes sont possibles. Cet attribut ne doit pas être omis. "yes" Il convient d'afficher la table de la recherche de mot vide. "no" sinon.
end:	Permet de préciser s'il convient d'afficher la table de recherche pour la recherche de fin de mot. Les valeurs suivantes sont possibles. Cet attribut ne doit pas être omis. "yes" Il convient d'afficher la table de la recherche de fin de mot. "no" sinon.
help_page_id:	Permet de préciser l'ID de page des données de flot liées à la table (explication de la table, par exemple). Peut être omis.

[Éléments-enfants]

<enable_key_type> Permet de définir le type des caractères qui peuvent être utilisés pour stocker la clé de recherche des mots indexés dans la table. Dans certaines langues comme l'anglais, la clé de recherche est simplement le mot lui-même. Toutefois, d'autres langues (le japonais, par exemple) peuvent utiliser la prononciation du mot à la place de sa représentation idéographique. Cet élément ne doit pas être omis. Ses attributs sont les suivants.

[Attributs]

numerals: Permet de définir si la clé de recherche peut ou non contenir des chiffres. Les valeurs possibles sont "yes" et "no". S'il est omis, la valeur par défaut est "no".

basic_alphabet: Permet de définir si la clé de recherche peut contenir l'alphabet (limité aux caractères non accentués de l'US-ASCII) ou non. Les valeurs possibles sont "yes" et "no". S'il est omis, la valeur par défaut est "no".

Voir A.2.6 sur la localisation des attributs spécifiques à une langue. Les jeux de caractères précédents sont définis dans le Tableau 5.

Tableau 5 – Caractères utilisables pour la clé de recherche

Nom du jeu de caractères	Caractères correspondants (toutes les valeurs sont au format Unicode)
numerals	numerals: 0 à 9 (0x0030 à 0x0039)
basic_alphabet	Alphabet US-ASCII standard: A à Z (0x0041 à 0x005A) et a à z (0x0061 à 0x007A)

<key_normalization> Permet de définir les méthodes de normalisation à appliquer sur les clés enregistrées dans cette table de recherche. Cet attribut ne doit pas être omis. L'interprétation de cette balise diffère en fonction des modes de recherche (recherche "correspondance uniquement" ("*matches-only*") et recherche "première correspondance" ("*matches-first*"). Le mode de recherche à utiliser est déterminé par l'attribut search_type de la balise <key_input_region> (voir 10.6.7 pour plus de détails).

a) recherche "correspondance uniquement"

(seuls s'affichent les mots indexés correspondant à l'input (élément entré); ils sont affinés à mesure que cet input est complété). L'input de l'utilisateur et les clés enregistrées dans la table de recherche sont normalisés selon les règles établies par l'attribut ci-dessous; la mise en correspondance est faite sur la forme normalisée.

b) recherche "première correspondance"

(les mots indexés s'affichent en commençant par ceux correspondant à l'input ou élément entré). Les paramètres sont ignorés, et la normalisation s'effectue selon les valeurs par défaut des attributs ci-dessous.

Les attributs ci-dessous sont disponibles.

[Attribut]

capitalization: Met tous les caractères (alphabétiques) en majuscules. Les valeurs possibles sont "yes" et "no". La valeur par défaut est "yes".

Voir la localisation (A.2.7) pour connaître les normalisations spécifiques à la langue.

Exemple:

```
<search_table>
  <search_table_def id="ST0001" use_default="yes">
    <enable_key_type numerals="no" basic_alphabet="yes"/>
    <key_normalization capitalization="yes"/>
  </search_table_def>
  <search_table_def id="ST0002">
    ...
  </search_table_def>
</search_table>
```

10.4 Module information-événement <event_info>

10.4.1 Événements

Dans la présente norme, la lecture sonore, les réactions aux clics, les liens de pages ou autres fonctions activées par l'utilisateur sont appelés événements. Le module information-événement enregistre les événements des données de flot. Son élément-enfant est le suivant.

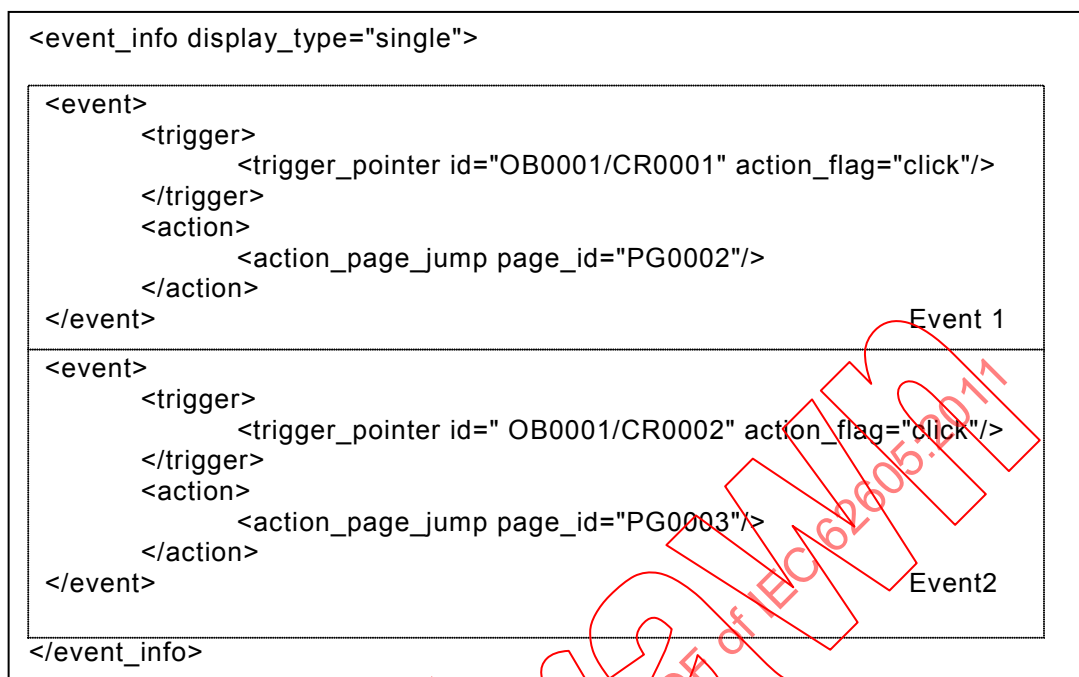
[Attribut]

display_type: Permet de spécifier les environnements des événements. Actuellement, seul "single" est possible. La valeur par défaut est "single" s'il est omis.

[Éléments-enfants]

<event> Données d'événement. Enregistre des informations relatives aux événements se produisant après l'affichage de l'objet pointé par l'attribut **body_id** de la balise <flow_data>, tels que des événements lancés par l'utilisateur. Par exemple, un clic sur une chaîne peut déclencher un saut vers une autre page. Voir 10.4.2 pour plus de détails. <event_info> peut contenir plusieurs instanciations de cette balise <event>. Si aucun événement ne doit être spécifié, la balise <event_info> elle-même doit être omise.

Exemple:



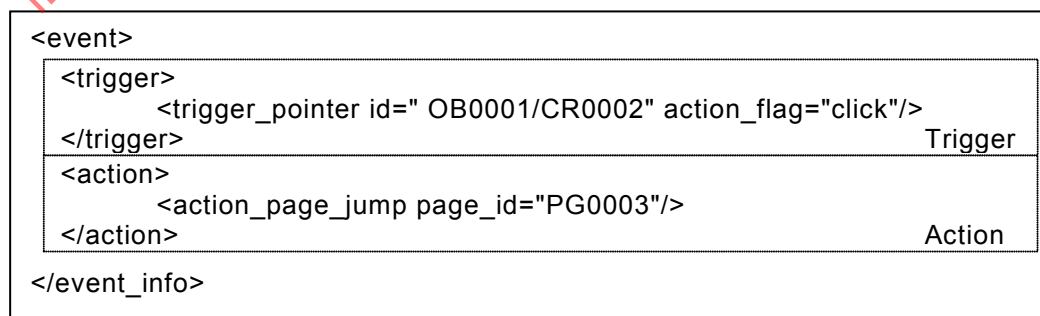
10.4.2 Données d'événement <event>

Un événement est composé de deux parties: le déclencheur et l'action. Le déclencheur est la condition qui déclenche l'événement, l'action décrivant ce qui doit être fait. Par exemple, des "clics de l'utilisateur dans une zone spécifique" ou une "pression sur un bouton" peuvent être des déclencheurs, l'action pouvant être un "saut vers une page particulière" ou la "lecture d'un fichier audio donné". Cette balise <event> comporte les éléments-enfants suivants.

[Éléments-enfants]

- <trigger> Voir 10.4.3 pour plus de détails. Il ne doit exister qu'une seule balise <trigger>, et elle ne peut comporter qu'un seul élément-enfant. Cet élément ne doit pas être omis.
- <action> Voir 10.4.4 pour plus de détails. Il ne doit exister qu'une seule balise <action>, et elle ne peut comporter qu'un seul élément-enfant. Cet élément ne doit pas être omis.

Exemple:



10.4.3 Déclencheur (pointeur) <trigger_pointer>

La balise <trigger> comporte les éléments-enfants suivants.

Définit des déclencheurs tels que des clics sur une zone. Ce <trigger_pointer> comporte les éléments-enfants et les attributs suivants.

[Attributs]

id: Permet de pointer vers la cible du déclencheur. Cet attribut ne doit pas être omis. Écrit sous la forme *Object_ID/Char_ID*. Ne pas confondre avec Reference ID (voir 10.6.3.4).

Exemple:

```
id="OB003k/CR0023"
```

La partie Char_ID de cette chaîne peut uniquement faire référence à ce qui suit (voir 10.6.2.3 pour plus de détails):

- a) le numéro d'ID défini dans l'attribut char_ID d'une balise <char_id>;
- b) le numéro d'ID défini dans l'attribut char_ID d'une balise <object>.

Toutefois, si une image hypertexte est définie à l'aide de la balise <pointer_region>, seul le deuxième cas (char_ID d'une balise <object>) peut être utilisé.

Noter que le numéro d'ID défini dans l'attribut déclencheur d'une balise <mask> (voir 10.6.2.3) ne doit pas être utilisé ici. S'il l'est, l'attribut déclencheur de la balise <mask> est ignoré.

action_flag: le type d'action qui active le déclencheur. Dans la norme actuelle, seul "click" est admis et utilisé par défaut en cas d'omission.

"click": un clic dans la zone cible.

[Éléments-enfants]

<pointer_region> Si une zone de déclencheur est limitée à une partie seulement de l'image vers laquelle pointe l'attribut id de la balise <trigger_pointer>, cette balise <pointer_region> sert à décrire une *Polygonal_region* (voir 9.15). Si l'attribut id de <trigger_pointer> ne pointe pas vers une image, le contenu de cet élément est ignoré. Peut être omis.

Si les événements comportent des déclencheurs qui se recouvrent, il peut y avoir une certaine ambiguïté quant à l'événement prioritaire en cas de clic sur les zones de recouvrement, en fonction du type de pointeur utilisé. Dans ce cas, la priorité est accordée aux événements dans l'ordre d'apparition à l'intérieur de <event_info>. L'exemple ci-dessous illustre cette situation.

Exemple:

```
<event_info>
  <event>
    <trigger>
      <trigger_pointer id="OB003k/CR0001" action_flag="click">
        <pointer_region>
          <vertex position="(0,0)"/>
          <vertex position="(100,0)"/>
          <vertex position="(100,100)"/>
          <vertex position="(0,100)"/>
        </pointer_region>
      </trigger_pointer>
    </trigger>
    <action>
      <action_page_jump page_id="PG0043"/>
    </action>
  </event>
  <event>
    <trigger>
      <trigger_pointer id="OB003k/CR0001" action_flag="click">
        <pointer_region>
          <vertex position="(50,50)"/>
          <vertex position="(100,50)"/>
          <vertex position="(100,100)"/>
          <vertex position="(50,100)"/>
        </pointer_region>
      </trigger_pointer>
    </trigger>
    <action>
      <action_page_jump page_id="PG0021"/>
    </action>
  </event>
</event_info>
```

Dans ce cas, les deux événements sont déclenchés en cliquant sur une sous-zone de OB003k/CR0001. Ces deux zones se chevauchent sur le carré (50, 50)-(100, 100). Un clic sur cette zone particulière doit déclencher le premier des deux événements, donnant lieu à un saut de page vers PG0043.

10.4.4 Informations d'action

10.4.4.1 Balise <action>

La balise <action> comporte les éléments-enfants suivants. Un seul d'entre eux peut être utilisé par balise <action>.

10.4.4.2 Action de lecture son <action_play>

Permet de lancer la lecture d'un son, d'une animation ou l'exécution d'autres morceaux qui peuvent être joués. Ses attributs sont les suivants.

[Attributs]

object_id: Permet de pointer vers l'objet à lire. Écrit comme indiqué ci-dessous. Cet attribut ne doit pas être omis.

Si l'objet est enregistré dans la table de gestion des objets (voir 10.5) (dans la présente version de cette norme, il s'agit uniquement des objets son et film), alors:

Écrit sous la forme *Object_ID*.

Exemple:

```
object id="OB003k"
```

Si l'objet est inséré dans l'*instanciation d'objet données de texte ou d'objet données de dictionnaire* par la balise <object> (dans la présente version de cette norme, possible uniquement pour les animations), alors:

Écrit sous la forme *Object_ID/Char_ID*.

Exemple:

```
object id="OB003k/CR0023"
```

Noter que la partie Char_ID doit être un numéro d'ID enregistré dans l'attribut char_id d'une balise <object> (10.6.2.3).

action: Permet de définir la méthode de lecture ou playback. La seule valeur acceptée est "normal". C'est aussi la valeur par défaut en cas d'omission. Si l'objet à exécuter est un élément sonore, et que le lecteur se déplace vers un autre flot ou atteint <page_break> ou </word> avec l'attribut de page_break à "yes" (uniquement dans les instanciations d'objets données de dictionnaire), la lecture du son doit être arrêtée. Dans le cas d'une animation, la lecture s'arrête quand il sort de l'écran.

Exemple:

```
<action_play object_id="OBkj23"/>
<action_play object_id="OB1234/CR0001"/>
<action_play object_id="OB1234/CR0001" action="normal"/>
```

10.4.4.3 Action de saut de page <action_page_jump>

Permet de passer de la page en cours à une autre page ou à un site Web. Ses attributs sont les suivants.

[Attributs]

book: Permet de définir le document cible du saut. S'il s'agit du même document que celui à l'origine du saut de page, il convient d'omettre cet attribut. Sinon, il y a trois types de saut indiqués ci-dessous:

a) Saut vers un site Web:

si la cible est un site Web html, il convient d'écrire l'attribut book comme une adresse URL commençant par http:// ou https:// (voir RFC 3986).

b) Appel téléphonique:

il convient d'écrire l'attribut book sous la forme "tel:" suivi du numéro de téléphone.

c) Écrire un message électronique à l'adresse indiquée:

il convient d'écrire l'attribut book sous la forme "mailto:" suivi d'une seule adresse électronique.

book_type: Utilisé pour différencier les types de contenu stockés dans l'attribut book. Doit être omis si l'attribut book l'est. Ne doit pas être omis si l'attribut book ne l'est pas.

a) Si l'attribut book pointe vers un site Web.

L'attribut doit contenir: "text/html"

b) Si l'attribut book pointe vers un numéro de téléphone.

L'attribut doit contenir: "application/x-tel"

c) Si l'attribut book pointe vers une adresse électronique.

L'attribut doit contenir: "application/x-mail"

page_id: Permet de définir les données de flot cible du saut, si la cible et l'origine du saut se trouvent dans le même document. Auquel cas, ne doit pas être omis. Écrit sous la forme *Page_ID*. Si l'attribut book est 1) un site Web, 2) un numéro de téléphone ou 3) une adresse électronique, cet attribut est omis.

center: Permet de préciser la destination du saut. Peut uniquement être utilisé si la cible du saut est un objet texte ou un objet données de dictionnaire (voir 10.6.2 et 10.6.3 pour plus de détails). Dans ce cas, la position à l'intérieur de ce texte est spécifiée à l'aide du Char_ID de la cible. Peut être omis. S'il est omis, la cible du saut est le début de l'ensemble de flot de données défini par l'attribut page_id. Si l'attribut book est 1) un site Web, 2) un numéro de téléphone ou 3) une adresse électronique, cet attribut est omis.

Exemple:

```
<action_page_jump page_id="PG23k4" />
<action_page_jump page_id="PG23k4" center="CR0001"/>
<action_page_jump book="http://www.sharp.co.jp/" book_type="text/html"/>
```

10.5 Module composantes <parts_module>

10.5.1 Stockage et gestion

Ce paragraphe stocke et gère de l'information sur les composantes des données de flot. L'élément <object_table> attribue un numéro d'ID objet et différents attributs à tous ces objets servant de contenu, par exemple des objets texte.

<parts_module> comporte l'élément-enfant suivant.

[Éléments-enfants]

<object_table> Module de gestion d'objet. Tous les objets utilisés dans le livre y sont enregistrés. Chaque objet est enregistré avec l'un des éléments décrits dans les paragraphes ci-dessous. Ne doit pas être omis.

[Éléments-enfants]

<dynamic_text_object_entry> Permet d'enregistrer des objets de texte servant de base pour le contenu courant.

<sound_object_entry> Permet d'enregistrer des objets son.

<search_page_object_entry> Permet d'enregistrer des objets page de recherche.

<movie_object_entry> Permet d'enregistrer des objets film.

<dict_data_object_entry> Permet d'enregistrer des objets données de dictionnaire.

Exemple:

```
<parts_module>
  <object_table>
    <dynamic_text_object_entry object_id="OBT001" ... >
      ...
    </dynamic_text_object_entry>
    <dynamic_text_object_entry object_id="OBT002" ... >
      ...
    </dynamic_text_object_entry>
    <sound_object_entry object_id="OBS001" ...>
      ...
    </sound_object_entry>
    <dynamic_text_object_entry object_id="OBT003" ... >
      ...
    </dynamic_text_object_entry>
    <search_page_object_entry object_id="OBSP01" ... >
      ...
    </search_page_object_entry>
    <movie_object_entry object_id="OBMV01" ... />
    ...
  </object_table>
</parts_module>
```

Les paragraphes ci-dessous expliquent comment chaque objet est enregistré.

10.5.2 Objet texte dynamique <dynamic_text_object_entry>

Permet d'enregistrer les objets texte à utiliser comme composantes du contenu courant. Cette balise comporte l'élément-enfant et les attributs suivants.

[Attributs]

- src:** Nom de fichier de l'*instanciation d'objet texte*. Ne doit pas être omis. Avant d'ouvrir le fichier indiqué dans cet attribut, il doit être comparé à l'attribut de type.
- type:** Permet de stocker le type MIME de l'*instanciation d'objet texte*. La seule valeur possible est "text/x-bvf-text". Ne doit pas être omis.
- object_id:** Permet d'attribuer un numéro d'ID à l'objet. Ce numéro est utilisé à partir des données de flot ou des modules de données événement, pour faire référence à cet objet. Écrit au format *object_ID* standard. Ne doit pas être omis.

[Éléments-enfants]

- <permission_info>** Permet de définir les autorisations pour l'objet sur lequel pointe l'attribut src. Écrit au format *Permission* standard. Peut être omis.

Exemple:

```
<dynamic_text_object_entry src="sect1.xml" type="text/x-bvf-text" object_id="OB03k2"/>
<dynamic_text_object_entry src="sect1.xml" type="text/x-bvf-text" object_id="OB03k1">
  <permission_info>
    <print_permission permission="authorized"/>
  </permission_info>
</dynamic_text_object_entry>
```

10.5.3 Objet son <sound_object_entry>

Permet d'enregistrer des objets son. Cette balise comporte les attributs et l'élément-enfant suivants.

[Attributs]

type: Permet d'enregistrer le type MIME de l'objet son. Dans la présente norme, les types suivants sont admis. Ne doit pas être omis

MP3:	"audio/mp3"
AAC:	"audio/3gpp2"
SMAF:	"application/x-smaf"

src: Nom de fichier de l'objet son, écrit au format *Filename* standard. Ne doit pas être omis. Avant d'ouvrir le fichier indiqué dans cet attribut, il doit être comparé à l'attribut de type. Actuellement, seul mp3 (extension .mp3), AAC (extension .3g2) et SMAF (extension .mmf) sont admis.

[Éléments-enfants]

<sound_object_info> Permet d'enregistrer les informations relatives à l'objet son. L'attribut ci-dessous est disponible.

[Attribut]

object_id: Permet d'attribuer un numéro d'ID à l'objet. Ce numéro est utilisé par le module données d'événement pour faire référence à cet objet. Écrit au format *object_ID* standard. Ne doit pas être omis.

Exemple:

```
<sound_object_entry src="foobar.mp3" type="audio/mp3">
  <sound_object_info object_id="OB143s"/>
</sound_object_entry>
```

10.5.4 Objet page de recherche <search_page_object_entry>

Permet d'enregistrer des objets page de recherche. Ses attributs et son élément-enfant sont les suivants.

[Attributs]

src: Nom de fichier de l'objet page de recherche, écrit au format *Filename* standard. Ne doit pas être omis. Avant d'ouvrir le fichier indiqué dans cet attribut, il doit être comparé à l'attribut de type.

- type:** Permet d'enregistrer le type MIME de l'objet page de recherche. Dans la présente spécification, seul "text/x-bvf-search-page" est admis. Ne doit pas être omis.
- object_id:** Permet d'attribuer un numéro d'ID à l'objet. Ce numéro est utilisé par le module données d'événement pour faire référence à cet objet. Écrit au format *object_ID* standard. Ne doit pas être omis.

[Éléments-enfants]

- <permission_info>** Permet de définir les autorisations relatives à l'objet vers lequel pointe l'attribut src. Écrit au format *Permission* standard. Peut être omis.

Exemple:

```
<search_page_object_entry
  src="spage1.xml" type="text/x-bvf-search-page" object_id="OBSP01" />

<search_page_object_entry
  src="spage2.xml" type="text/x-bvf-search-page" object_id="OBSP02" >
  <permission_info>
    <print_permission permission="authorized"/>
  </permission_info>
</search_page_object_entry>
```

10.5.5 Objet film <movie_object_entry>

Enregistre des objets film. Ses attributs et son élément-enfant sont les suivants.

[Attributs]

- src:** Nom de fichier de l'objet film, écrit au format *Filename* standard. Ne doit pas être omis. Avant d'ouvrir le fichier indiqué dans cet attribut, il doit être comparé à l'attribut de type. Actuellement, seul 3GPP2 (extension .3g2) est admis.
- type:** Permet d'enregistrer le type MIME de l'objet film. Dans la présente norme, le type suivant est admis. Ne doit pas être omis.
3GPP2: "video/3gpp2"
- object_id:** Permet d'attribuer le numéro d'ID à l'objet. Ce numéro est utilisé par le module données d'événement pour faire référence à cet objet. Écrit au format *object_ID* standard. Ne doit pas être omis.

Exemple:

```
<movie_object_entry src="movie1.3g2" type="video/3gpp2" object_id="OBmv00"/>
```

10.5.6 Objet données de dictionnaire <dict_data_object_entry>

Cette balise enregistre les objets données de dictionnaire de texte. Cette balise comporte l'élément-enfant et les attributs suivants.

[Attributs]

- src:** Nom de fichier de l'*instance d'objet données de dictionnaire*. Ne doit pas être omis. Avant d'ouvrir le fichier indiqué dans cet attribut, il doit être comparé à l'attribut de type.

- type:** Permet de stocker le type MIME de l'*instanciation d'objet données de dictionnaire*. La seule valeur possible est "text/x-bvf-dict-data". Ne doit pas être omis.
- object_id:** Permet d'attribuer le numéro d'ID à l'objet. Ce numéro est utilisé à partir des données de flot ou des modules données d'événement pour faire référence à cet objet. Écrit au format *object_ID* standard. Ne doit pas être omis.

[Éléments-enfants]

<permission_info> Permet de définir les autorisations pour l'objet sur lequel pointe l'attribut src. Écrit au format *Permission* standard. Peut être omis.

Exemple:

```
<parts_module>
  <object_table>
    <dict_data_object_entry src="dict1.xml" type="text/x-bvf-dict-data" object_id="OBD001">
      <permission_info>...</permission_info>
    </dict_data_object_entry>
    ...
  </object_table>
</parts_module>
```

10.6 Instanciation d'un objet

10.6.1 Généralités

Dans la présente norme, le terme *instanciation d'un objet* est utilisé pour faire référence aux objets que le dispositif de visualisation affiche (ou lit, selon le cas). Les types possibles d'*instanciation d'objet* sont indiqués ci-dessous:

Instanciation d'objet texte

Instanciation d'objet données de dictionnaire

Instanciation d'objet image

Instanciation d'objet son

Instanciation d'objet animation

Instanciation d'objet page de recherche

Instanciation d'objet film

10.6.2 Instanciation d'objet texte <text_data>

10.6.2.1 Instanciations d'objet texte

Les *instanciations d'objet texte* sont stockées dans leurs propres fichiers XML, l'élément racine étant <text_data> (voir ci-dessous).

Exemple:

```
<?xml version="1.0" encoding="UTF-8" ?>

<text_data>
  <text_default_attribute>
    ...
  </text_default_attribute>
  <text_body>
    Two households, both alike in dignity, ...
  </text_body>
</text_data>
```

Comme illustré, l'élément `<text_data>` comporte deux éléments-enfants principaux, l'un pour stocker les attributs par défaut du texte (`<text_default_attribute>`) et l'autre pour stocker le texte lui-même (`<text_body>`).

10.6.2.2 Attributs par défaut du texte `<text_default_attribute>`

Les attributs par défaut du texte stockent des paramètres à utiliser lors de l'affichage du texte. A noter que ces paramètres peuvent ne pas toujours être respectés si le dispositif de visualisation n'est pas en mesure de traiter les valeurs/comportements établis dans ce module ou si le dispositif de visualisation est configuré pour donner la priorité aux paramètres utilisateur. En outre, comme cela a été expliqué en 10.3.2.3, certains paramètres présentés ici peuvent également être définis dans l'ensemble du flot de texte (text flow) (dans l'élément-enfant `<flow_default_attribute>` de `<flow_entry>`. Voir 10.3.2.4). Ils sont utilisés par défaut si aucune valeur n'a été attribuée aux paramètres locaux (ceux définis ici). S'ils sont tous deux définis, la priorité est accordée aux paramètres locaux.

La balise `<text_default_attribute>` comporte les attributs et éléments-enfants suivants.

[Attributs]

- baseline:** Permet de définir l'orientation de la ligne de base (et donc du texte) pour chaque flot. Les valeurs possibles sont indiquées ci-dessous. S'il est omis, la valeur par défaut est `<flow_default_attribute>`.
- "right": Le sens d'écriture est horizontal (de gauche à droite). Toutefois, il peut être modifié par l'utilisateur.
 - "right_only": Le sens d'écriture est horizontal (de gauche à droite), et l'utilisateur ne peut pas modifier le paramètre. Toutefois, s'il n'est pas pris en charge par le dispositif de visualisation, ce paramètre n'est pas nécessairement appliqué.
 - "down": Le sens d'écriture est vertical (de haut en bas). Toutefois, il peut être modifié par l'utilisateur.
 - "down_only": Le sens d'écriture est vertical (de haut en bas), et l'utilisateur ne peut pas modifier le paramètre. Toutefois, s'il n'est pas pris en charge par le dispositif de visualisation, ce paramètre n'est pas nécessairement appliqué.
- valign:** Permet de déterminer la manière de placer la zone de texte. Ici sont déterminées la position verticale pour du texte écrit horizontalement ou la position horizontale pour du texte écrit verticalement. La valeur décrite ci-dessous remplit les conditions requises. Si elle est omise, il convient d'afficher la zone de texte à partir du haut de la zone d'affichage.
- "middle": Le contenu de la balise `<text_body>` est centré au milieu de la zone d'affichage (voir ci-dessous). Si le contenu est

plus grand que la zone d'affichage, cet attribut est ignoré, voir Figure 5.

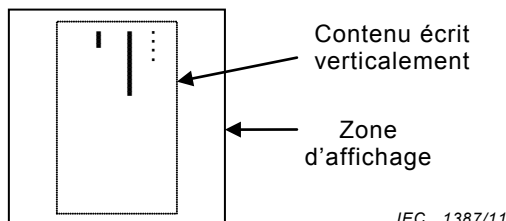


Figure 5 – Exemple de valign="middle"

[Éléments-enfants]

<text_default_font> Permet de définir la police par défaut et sa couleur pour cet objet texte. S'il est omis, la valeur par défaut est <flow_default_attribute>. Ses attributs sont les suivants.

[Attributs]

fontname: Nom de la police par défaut. Plusieurs polices peuvent être spécifiées. Dans ce cas, il convient de les séparer par une virgule (0x2c en Unicode). Par exemple:

fontname="Aaa sans serif,Bbb gothic"

Le dispositif de visualisation utilisera la première police répertoriée disponible. S'il est omis, la valeur par défaut est <flow_default_attribute>.

color_space, color, opacity:

Permet de définir la couleur de police à utiliser pour les données de contenu. Écrit au format de données *color* standard. S'il est omis, la valeur par défaut est <flow_default_attribute>.

ruby_flag:

Permet de définir si le code ruby des données de contenu doit être visualisé ou non. S'il est omis, la valeur par défaut est <flow_default_attribute>. Le texte à afficher lorsque le code ruby est actif est celui placé dans la balise <ruby> des instanciations d'objet texte. Les valeurs suivantes peuvent être utilisées.

"yes": Il convient d'afficher le code ruby, mais peut être désactivé par l'utilisateur.

"yes_only": Le code ruby doit être affiché et ne peut pas être désactivé par l'utilisateur. Toutefois, cela ne concerne pas les dispositifs de visualisation qui ne sont pas en mesure d'afficher le code ruby.

"no": Il est recommandé de ne pas afficher le code ruby, mais il peut toujours être activé en fonction des préférences de l'utilisateur.

"no_only": Le code ruby ne doit pas être affiché et ne peut pas être désactivé par l'utilisateur. Toutefois, cela ne s'applique pas si le dispositif de visualisation est incapable de désactiver l'affichage du code ruby.

<text_default_background> Permet de définir la couleur et l'image d'arrière-plan à utiliser pour cet objet texte. Si ces deux éléments sont définis, l'image est dessinée centrée dans l'écran avec la couleur d'arrière-plan. Enfin, le contenu de la balise **<text_body>** est rendu en haut. Si l'image d'arrière-plan est trop grande pour tenir dans l'écran, elle est réduite et mise à l'échelle en préservant le rapport hauteur/largeur. Si cette balise est omise il convient que le comportement soit celui spécifié lorsque ses attributs sont omis. Les attributs et éléments-enfants ci-dessous sont disponibles.

[Attributs]

color_space, color, opacity:

Permet de définir la couleur à utiliser en arrière-plan. Écrit au format de données *color* standard. S'il est omis, la valeur par défaut est **<flow_default_attribute>**.

type: Permet de stocker le type MIME de l'image d'arrière-plan. Peut être omis uniquement si l'attribut **src** est aussi omis. Les valeurs possibles sont indiquées ci-dessous.

"image/png"

"image/jpeg"

src: Permet de définir le nom de fichier de l'image à utiliser en arrière-plan, écrit au format *Filename* standard. Peut être omis. Avant d'ouvrir le fichier indiqué dans cet attribut, il doit être comparé à l'attribut de type.

[Éléments-enfants]

<permission_info>:

Permet de définir les autorisations pour l'image à laquelle fait référence l'attribut **src**. Écrit au format *Permission* standard. Peut être omis. Si l'attribut **src** est omis, il convient d'ignorer cette information d'autorisation.

<text_default_background_music>

Permet de définir la musique d'arrière-plan à jouer lors de l'affichage de cet objet texte. Si cette balise est omise, il convient de ne rien jouer. Les attributs et éléments-enfants ci-dessous sont disponibles.

[Attributs]

type: Permet de stocker le type MIME de la musique d'arrière-plan. Cet attribut ne doit pas être omis. Les valeurs possibles sont indiquées ci-dessous.

"audio/mp3"

"audio/3gpp2"

"application/x-smaf"

src: Permet de définir le nom de fichier de la musique d'arrière-plan, écrit au format *Filename* standard. Cet attribut ne doit pas être omis. Avant d'ouvrir le fichier indiqué dans cet attribut, on doit le comparer à l'attribut de type.

loop: Permet d'indiquer s'il convient de jouer la musique d'arrière-plan de manière itérative (c'est-à-dire répétée depuis le début à chaque fois que la fin est atteinte). Les valeurs possibles sont "yes" et "no". En cas d'omission, la valeur par défaut est no.

[Éléments-enfants]

<permission_info>:

Permet de définir les autorisations pour le fichier son auquel fait référence l'attribut src. Écrit au format *Permission* standard. Peut être omis.

<line_breaking_method> Spécifie l'algorithme à utiliser pour déterminer la manière de couper le texte en lignes. Les différentes langues traitent cela de diverses manières. Cet élément autorise une certaine souplesse. S'il est omis, la valeur par défaut est <flow_default_attribute>. Cette balise comporte les attributs et éléments-enfants suivants.

[Attributs]

method: Permet de choisir la méthode de saut de ligne. La valeur par défaut est "none". Les valeurs possibles sont:

"none": aucun traitement particulier. Si une ligne est pleine de caractères, passer à la suivante.

D'autres valeurs sont possibles pour la localisation linguistique. Voir Article A.2.

D'autres attributs peuvent être ajoutés par des méthodes localisées. Voir Article A.2.

[Éléments-enfants]

Aucun, à l'exception de ceux ajoutés par des méthodes localisées. Voir Article A.2.

Exemple:

```
<text_default_attribute>
  <text_default_font fontname="foobar-serif" color="green"/>
  <text_default_background color="black" src="001.jpg" type="image/jpeg"/>
  <text_default_background_music src="1.mp3" type="application/x-smaf"
loop="yes"/>
  <line_breaking_method ="none">
  </line_breaking_method >
</text_default_attribute>
```

10.6.2.3 Corps du texte <text_body>

Cette partie permet d'enregistrer le corps du texte réel de l'objet texte. Dans cette balise, la *chaîne de caractères étendus*, ainsi que les éléments décrits ci-dessous, peuvent être utilisés.

a) Paragraphe

<p>...</p> Permet de créer un paragraphe. En règle générale, un saut de ligne est placé en début de paragraphe (immédiatement avant la balise <p>) et à la fin (immédiatement après la balise </p>). Toutefois, aucun saut de ligne n'est placé dans les cas énumérés ci-dessous:

si le paragraphe commence au début d'une ligne, même sans insertion de saut de ligne;

si ce qui suit le paragraphe commence au début d'une ligne, même sans insertion de saut de ligne.

Il convient également de noter que si la fin du paragraphe se termine dans un texte qui court autour d'une image (voir l'attribut align de la balise <object>), le saut de ligne n'annule pas cet effet. Par conséquent, la ligne suivante entoure encore l'image. Pour supprimer cet effet et commencer la nouvelle ligne après l'image, <br clear="all"/> doit être explicitement utilisé. Les attributs et éléments-enfants de <p> sont les suivants.

[Attributs]

top_line_indent: Permet de définir la taille (exprimée en em) de l'indentation de la première ligne du paragraphe. Les espaces sont insérés avant le premier caractère qui suit la balise <p> et le premier caractère qui suit une balise
 dans le paragraphe. L'unité est "em". L'espacement réel est la somme des valeurs définies dans cet attribut et la marge du paragraphe définie par l'attribut "top". De plus, une valeur négative peut être affectée à cet attribut ("-2em", par exemple). S'il est omis, la valeur par défaut est 0.

top: Permet de définir la taille de la marge de gauche du paragraphe (la marge supérieure si le sens d'écriture est vertical). Elle est exprimée en em ou en pourcentage de la longueur de la ligne (ou de la colonne si le sens d'écriture est vertical). Si elle est omise, la valeur par défaut est "0em". Des valeurs négatives ne doivent pas être utilisées. Des paragraphes peuvent être contenus dans un autre paragraphe. Dans ce cas, les marges s'ajoutent. Les pourcentages sont également en rapport avec la zone interne calculée, et non avec la zone d'affichage totale.

bottom: Permet de définir la taille de la marge de droite du paragraphe (la marge inférieure si le sens d'écriture est vertical). Elle est exprimée en em ou en pourcentage de la longueur totale de la ligne (de la colonne si le sens d'écriture est vertical). Si elle est omise, la valeur par défaut est "0em". Les valeurs négatives ne doivent pas être utilisées. Des paragraphes peuvent être contenus dans un autre paragraphe. Dans ce cas, les marges s'ajoutent. Les pourcentages se rapportent à la zone interne ainsi calculée, et non à la zone d'affichage totale.

align: Permet de déterminer si le texte doit être au milieu ou aligné à droite ou aligné à gauche, en respectant la longueur de ligne définie par les attributs d'indentation pour le haut, le bas et la première ligne. Cet attribut accepte les valeurs énumérées ci-dessous. S'il est omis, le paramètre en cours ne change pas.

"center": La chaîne placée dans la balise <p> s'affiche au centre.

"right": La chaîne placée dans la balise <p> est alignée à droite (en bas en cas d'écriture verticale).

"left": La chaîne placée dans la balise <p> est alignée à gauche (en haut en cas d'écriture verticale).

drop_cap: Permet de remplacer la première lettre du paragraphe par une lettrine. Sa valeur est un entier indiquant le nombre

de lignes recouvertes par la lettrine. La valeur par défaut est "1" (taille normale); voir Figure 6.

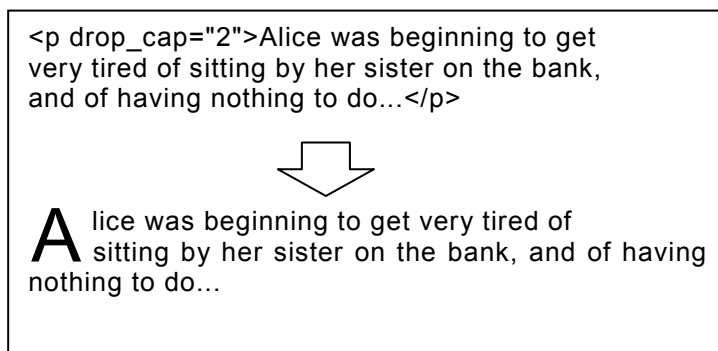


Figure 6 – Exemple de lettrine

[Éléments-enfants]

A l'exception des balises `<page_break/>` et `<head>`, tous les éléments qui peuvent être placés dans `<text_body>`, ainsi que les *chaînes de caractères étendus*, peuvent figurer dans le paragraphe balisé par `<p>`.

La Figure 7 illustre un exemple de paragraphe rédigé horizontalement avec `top="25%"` et `bottom="2em"`.

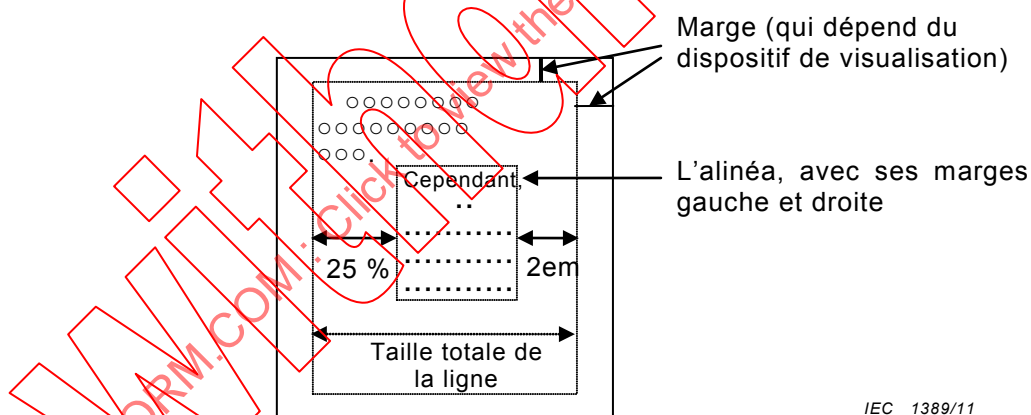


Figure 7 – Marges de gauche et de droite d'un paragraphe

Exemple:

```
<p>.... </p>
<p align="center"> ... </p>
<p top="25%" bottom="2em"> ... </p>
```

`<scrolling_text>` Permet de définir une ligne de texte déroulant. Les sauts de ligne sont automatiquement insérés autour de cet élément (immédiatement avant la balise `<scrolling_text>` et après la balise `</scrolling_text>`) pour s'assurer que le texte déroulant est seul sur sa ligne. La hauteur de l'objet texte déroulant dépend entièrement de son contenu. Pour un texte écrit horizontalement, le déroulement se fait de droite à gauche tandis que pour

un texte écrit verticalement, il convient que le déroulement déplace le texte vers le haut, à partir du bas. Il convient que le texte soit déroulé jusqu'à ce qu'il disparaisse de la zone d'affichage limitée par les marges en haut et en bas (marges "top" et "bottom") du paragraphe environnant, s'il y en a un, et ensuite qu'il redémarre. La balise `<scrolling_text>` accepte les éléments-enfants suivants.

[Éléments-enfants]

La chaîne à afficher comme texte déroulant est écrite comme un élément-enfant, sous la forme d'une *chaîne de caractères étendus*. Les balises suivantes peuvent également être utilisées: `<external_char>`, `<font>`, `<horizontal>`, `<ruby>`, `<sub>`, `<sup>` et `<object>`. Toutefois, la balise `<object>` fait l'objet des limitations supplémentaires suivantes lorsqu'elle est utilisée comme élément-enfant de la balise `<scrolling_text>`:

seuls des fichiers image peuvent être désignés par l'attribut `src`,

seule une valeur "top", "middle" ou "bottom" peut être affectée à l'attribut `align`;

l'attribut `char_id` ne doit pas être utilisé.

Exemple:

```
<scrolling_text>
  This text will be scrolling over and over
</scrolling_text>
```

b) Éléments en-ligne

`<br/>`

Permet d'insérer un saut de ligne. Cette balise ne doit pas comporter d'élément-enfant.

[Attribut]

`clear`:

Permet d'annuler le texte courant autour d'un objet intégré tel qu'une image. La balise `<br/>` ne procède pas à l'annulation si cet attribut est omis. Les valeurs suivantes sont possibles:

"left":

Permet d'insérer des lignes en blanc après la balise `<br/>` jusqu'à ce que le texte puisse être inséré au début de la ligne (gauche pour l'écriture horizontale et haut pour l'écriture verticale). En d'autres termes, il déplace le point d'insertion après l'objet autour duquel court le texte, s'il y a un tel objet, s'il se trouve à gauche.

"right":

Permet d'insérer des lignes en blanc après la balise `<br/>` jusqu'à ce que le texte puisse être inséré au début de la ligne (droite pour l'écriture horizontale, et en bas pour l'écriture verticale). En d'autres termes, il déplace le point d'insertion après l'objet, autour duquel court le texte, s'il y a un tel objet, s'il se trouve à droite.

"all":

Permet d'insérer des lignes en blanc après la balise `<br/>` jusqu'à ce que le texte puisse être inséré du début jusqu'à la fin de la ligne. En d'autres termes, il déplace le point d'insertion après l'objet,

autour duquel court le texte, s'il y a un tel objet, qu'il se trouve sur la gauche ou sur la droite.

Exemple:

```
<br/>
<br clear="all"/>
```

<hr/>

Trace une ligne horizontale (ou verticale si le texte est écrit verticalement). Comme cette ligne doit toujours être unique dans sa rangée (ou sa colonne si le texte est vertical), elle est toujours suivie, et précédée aussi, d'un saut de ligne (équivalent à
), à moins que la ligne ne commence à partir du haut de son propre chef. Ses attributs sont les suivants, et elle ne comporte aucun élément-enfant.

[Attributs]

size: Permet de déterminer l'épaisseur de la ligne. Exprimée en "em". Noter que cette valeur n'a pas à être un entier. S'il est omis, sa valeur par défaut est "0,1em".

length: Permet de déterminer la longueur de la ligne. Exprimé en "em" ou en pourcentage de l'espace disponible (en ligne si le texte est écrit horizontalement, en colonne s'il est écrit verticalement), en considérant les attributs "top" et "bottom" de la balise <p> qui délimite ce paragraphe. S'il est omis, sa valeur par défaut est "100%".

align: Permet de définir la position de la ligne si elle est plus courte (selon l'attribut length) que la zone disponible. Les valeurs possibles sont "left", "center" et "right". S'il est omis, la valeur par défaut est "center".

Exemple:

```
<hr/>
<hr size="0.5em" length="50%" align="center"/>
```

...

Permet de définir les propriétés de la police pour le texte placé entre les balises. Ses attributs et éléments-enfants sont les suivants.

[Attributs]

name: Permet de définir la police à utiliser, en indiquant son nom. S'il est omis, la police utilisée avant la balise s'applique encore. Plusieurs polices peuvent être spécifiées. Dans ce cas, les noms des polices sont séparés par une virgule (0x2c en Unicode). Par exemple:

fontname="Aaa sans serif,Bbb gothic"

Le dispositif de visualisation utilisera la première police répertoriée disponible.

size: Permet de définir une taille de police différente. La taille est définie en pourcentage de celle définie par l'attribut de base. S'il est omis, la taille de base n'est pas modifiée.

base: Permet de définir la base du calcul de l'attribut size. Les valeurs indiquées ci-dessous sont acceptées. S'il est omis, sa valeur par défaut est "last".

"last":	Taille identique à celle du caractère qui précède la balise .
"default":	Taille identique à la taille de police par défaut.
color_space, color, opacity:	Permet de définir la couleur du caractère. Écrit au format de données <i>Color</i> standard (voir 9.16 pour plus de détails). S'il est omis, la couleur des caractères précédant immédiatement la balise est utilisée.
bold:	Permet de définir s'il convient d'afficher en gras la chaîne incluse dans la balise . Les valeurs admises sont: "yes": afficher en gras "no": afficher normalement S'il est omis, l'état en cours est maintenu.
underline:	Permet de définir s'il convient de souligner la chaîne incluse dans la balise . "yes": afficher en soulignant "no": afficher normalement Notons que les objets (par exemple objets images) compris entre les balises ne doivent pas être soulignés, même si cet attribut a la valeur "yes". S'il est omis, l'état en cours est maintenu.
italic:	Permet de décider si les données de contenu sont à afficher en italique ou non. Si cet attribut est omis, le comportement dépend du dispositif de visualisation. Les valeurs admises sont: "yes": afficher en italique "no": afficher normalement S'il est omis, l'état en cours est maintenu.
oblique:	Permet de décider d'afficher en oblique les données de contenu. Si cet attribut est omis, le comportement dépend du dispositif de visualisation. Les valeurs admises sont: "yes": afficher en oblique "no": afficher normalement S'il est omis, l'état en cours est maintenu.
small_caps:	Permet de décider d'afficher les données de contenu en petites capitales. Si cet attribut est omis, le comportement dépend du dispositif de visualisation. Les valeurs admises sont: "yes": afficher en petites capitales "no": afficher normalement S'il est omis, l'état en cours est maintenu.
family:	Permet de spécifier la famille de polices à utiliser. La police effectivement utilisée pour le rendu dépend du dispositif de visualisation. Si cet attribut est omis, le comportement dépend encore du dispositif de visualisation. Les valeurs admises sont: "monospace": afficher en polices de largeur fixe "sans-serif": afficher en polices sans serifs "serif" afficher en polices avec serifs "cursive" afficher en polices de style manuscrit

"fantasy" afficher en polices décoratives

S'il est omis, l'état en cours est maintenu.

ul_type Permet de spécifier la manière dont les chaînes marquées par la balise <u> et son équivalent sont rendues. Les valeurs acceptables sont les suivantes, toutes issues de l'ISO/CEI 9541-1 (8.7.1.12.1.1 Score Name). Voir cette norme pour connaître le sens exact de ces valeurs.

"rightscore"

"leftscore"

"throughscore"

em_type: Permet de spécifier la manière dont les chaînes marquées par la balise et son équivalent sont rendues. Les valeurs acceptables se présentent comme suit, dont les quatre premières sont issues de l'ISO/CEI 9541-1: (8.7.1.12.1.1 Score Name). Voir ce document pour connaître le sens exact de ces valeurs.

Noter que lorsque cet attribut est spécifié, la balise n'est plus équivalente à

"rightscore"

"leftscore"

"throughscore"

"kendot"

bold: Il convient de rendre les sous-chaînes en gras.

italic: Il convient de rendre les sous-chaînes en italique.

bold_italic: Il convient de rendre les sous-chaînes en gras italique.

"reverse" Il convient de rendre la sous-chaîne avec les caractères et l'arrière-plan inversés.

"shade" Il convient de rendre les sous-chaînes avec une ombre.

[Éléments-enfants]

Toutes les combinaisons de *chaînes de caractères étendus* et tous les éléments en ligne (à l'exception de <page_break/>) et <object> peuvent être utilisés.

Exemple:

```
<font name="Aaa sans serif">.....</font>
<font size="200%">.....</font>
```

<i>...</i> Permet de spécifier les sous-chaînes à de présenter en italique. Equivalent à ...

<u>...</u> Permet de spécifier les sous-chaînes à présenter soulignées. Equivalent à Son attribut est le suivant.

ul_type Permet de spécifier le rendu des chaînes marquées par la balise <u> et son équivalent. Les valeurs acceptables sont les suivantes, toutes issues de l'ISO/CEI 9541-1

(8.7.1.12.1.1 Score Name). Voir cette norme pour connaître le sens exact de ces valeurs.

Les valeurs acceptables sont issues de l'ISO/CEI 9541-1 et sont les suivantes:

"rightscore"

"leftscore"

"throughscore"

... Permet de spécifier les sous-chaînes à rendre en italique gras. Equivalent à **... ** dans les paramètres par défaut. Si l'attribut **em_type** de **<flow_default_font>** et **<text_default_font>** est spécifié, ce n'est plus le cas et le rendu des chaînes marquées par cette balise dépend de **em_type**. Les valeurs acceptables se présentent comme suit, les quatre premières issues de l'ISO/CEI 9541-1 (8.7.1.12.1.1 Score Name). Voir ce document pour connaître le sens exact de ces valeurs.

em_type: Permet de spécifier la manière dont les chaînes marquées par la balise **** et son équivalent sont rendues. Peut être omis. Si **em_type** est spécifié, il remplace ce qui est indiqué par l'attribut **em_type** dans **<flow_default_font>**. Les valeurs acceptables se présentent comme suit, dont les quatre premières issues de l'ISO/CEI 9541-1.

"rightscore"

"leftscore"

"throughscore"

"kendot"

"bold": Il convient de rendre les sous-chaînes en gras.

"italic": Il convient de rendre les sous-chaînes en italique.

"bold_italic": Il convient de rendre les sous-chaînes en gras italique.

"reverse" Il convient de rendre la sous-chaîne avec les caractères et l'arrière-plan inversés.

"shade" Il convient de rendre les sous-chaînes avec une ombre.

<oblq>...</oblq> Permet de spécifier une sous-chaîne à rendre en oblique. Equivalent à **...**

<sc>...</sc> Permet de spécifier une sous-chaîne à rendre en petites capitales. Equivalent à **...**

<mlg>...</mlg> Permet de représenter une note multiligne. La présente norme n'indique pas de quelle façon il convient de présenter les notes multilignes; cependant, pour permettre une meilleure compréhension du concept, deux exemples de la façon dont elles doivent être présentées sont illustrés ci-dessous.

Exemple:

これが割注<mlg>わりちゅう</mlg>です。



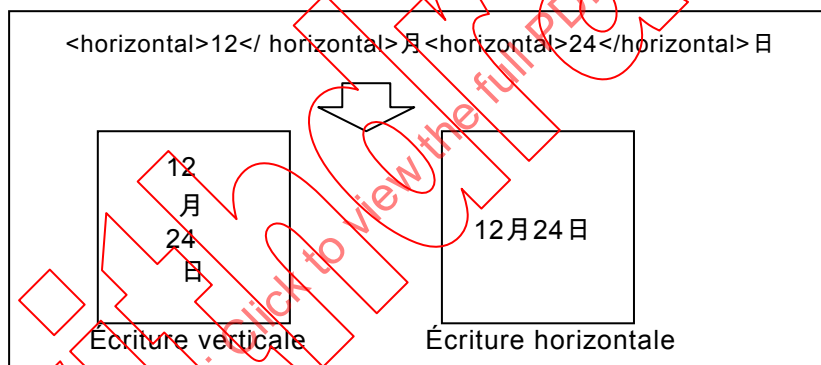
これが割注<sup>わり
ちゅう</sup>です。

これが割注 (わりちゅう)です。

<horizontal>...</horizontal> Permet de définir les sous-chaînes qu'il convient d'afficher horizontalement, même si la direction générale du texte est vertical. Il est souvent utilisé pour les dates et les nombres en japonais; voir la Figure 8. Si le texte est écrit horizontalement, cette balise n'a aucun effet. L'élément-enfant suivant est accepté.

[Éléments-enfants]

Les chaînes de caractères externes étendus peuvent être utilisées ici.



IEC 1390/11

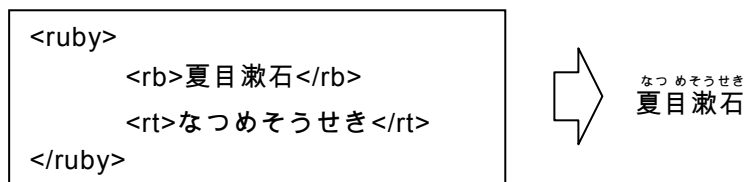
Figure 8 – Écriture horizontale dans un texte vertical

<ruby>...</ruby> Permet d'afficher le texte ruby. Ruby est une fonction souvent utilisée dans la typographie asiatique, et notamment japonaise. Il s'agit d'un court texte qui s'affiche au-dessus du texte de base, la plupart du temps utilisé comme guide de prononciation des caractères idéographiques. Dans la mesure où les polices disponibles le permettent, le texte ruby s'affiche habituellement en caractères plus petits que le texte de base qu'il annote. Le traitement de l'espacement pour ouvrir suffisamment d'espace pour afficher le texte ruby dépend du dispositif de visualisation. Il convient également de noter qu'étant donné qu'un texte ruby est une aide à la lecture et que certaines personnes peuvent souhaiter s'en dispenser ou ne pas en avoir besoin, certains dispositifs de visualisation peuvent proposer de désactiver complètement l'affichage du texte ruby, voir Figure 9. La balise <ruby> comporte les éléments-enfants suivants.

[Éléments-enfants]

- <rb> Chaîne de base qu'il convient d'annoter d'un texte ruby. Cet attribut ne doit pas être omis. Composé d'une chaîne de caractères externes étendus.
- <rt> La chaîne ruby qui s'affiche au-dessus de la base (ou à sa droite, si le texte s'affiche verticalement). Cet attribut ne

doit pas être omis. Composé d'une *chaîne de caractères externes étendus*.



IEC 1391/11

Figure 9 – Ruby

`<sub>...</sub>` Permet de représenter la sous-chaîne en indice.

`<sup>...</sup>` Permet de représenter la sous-chaîne en exposant.

`<external_char>` Permet d'insérer un *caractère externe*. La syntaxe détaillée est décrite en 9.11. Dans l'*instanciation d'objet texte*, les attributs `alt_img` et `alt_vimg` peuvent être utilisés.

Exemple:

```
<external_char alt_set="ooextchars" alt_code="0x1234"
  alt_img="ou.img" alt_vimg="ou_v.img"
  img_type="image/jpeg" alt="鷗"/>
```

`<mask>` Permet d'ajouter une capacité de masquage à la chaîne intégrée dans cette balise. En cas de clic sur cette chaîne ou celle désignée par l'attribut `trigger`, l'élément-enfant est couvert, puis découvert en alternance. Ses attributs et Éléments-enfants sont les suivants.

[Attributs]

`initial_flag`:

Permet de déterminer l'état initial du masque. Les valeurs possibles sont indiquées ci-dessous, la valeur par défaut étant "on" en cas d'omission.

"on": Le contenu est masqué.

"off": Le contenu n'est pas masqué.

`trigger`:

Permet de désigner la/les chaîne(s) qui déclenchent le changement d'état du masque. En cas de clic sur une chaîne indiquée par cet attribut, le masque bascule entre on et off. Écrit sous la forme d'un ou de plusieurs *Char_ID* séparés, dans ce dernier cas, par une virgule (0x2C en Unicode), comme indiqué ci-dessous.

Exemple:

"CR0001,CR0002,CR0003"

La chaîne de caractères indiquée doit se trouver dans la même *instanciation d'objet texte*. Noter que le `char_id` utilisé ici ne doit pas être spécifié en même temps qu'un attribut `trigger` dans le module informations d'événement. Si c'est le cas, aucune action n'est déclenchée sur le

masque, même en cas de clic sur la chaîne indiquée. S'il est omis, la chaîne placée dans le masque lui-même fait office de déclencheur.

char_id: Permet de définir un ID chaîne pour cet objet <mask>. Écrit au format *Char_ID* standard. Peut être omis.

mask_type: Permet de sélectionner le type/style du masque à utiliser. Les valeurs indiquées ci-dessous peuvent être utilisées. S'il est omis, la valeur par défaut est "default".

"default": L'apparence du masque dépend du dispositif de visualisation.

"color": Le masque est une zone remplie et colorée. La couleur est définie par les attributs correspondants ou par leur valeur par défaut en cas d'omission.

color_space, color, opacity: Permet de définir la couleur du masque, en utilisant le format *Color* standard (voir 9.16). Doit être défini uniquement si la valeur "color" est affectée à l'attribut mask_type. S'il est omis, la valeur par défaut est "black".

hold_flag: Permet de définir les règles d'enregistrement de l'état, permettant de se rappeler si le masque a été activé ou désactivé. Les valeurs ci-dessous peuvent lui être attribuées. S'il est omis, la valeur par défaut est "scope".

"scope": Permet de se rappeler l'état tant que l'instanciation d'objet texte s'affiche.

"on_power": Permet de se rappeler l'état tant que le document est ouvert.

"save": Il convient de sauvegarder l'état lors de la fermeture du document et de le restaurer au moment de sa réouverture.

[Éléments-enfants]

Le contenu de cet élément peut être toute combinaison de *chaînes de caractères externes étendus* et de balises
, , <horizontal>, <ruby> et <object>. Toutefois, les restrictions suivantes s'appliquent à la balise <object>:

L'attribut src doit pointer sur un fichier d'image.

L'attribut char_id ne doit pas être utilisé.

Exemple:

The answer is <mask>42 </mask>.
The answer is <mask trigger ="CR0002">42</mask>.

<char_id> </char_id> Permet d'associer un numéro d'ID de chaîne à son contenu. Son attribut et ses éléments-enfants sont les suivants.

[Attribut]

char_id: Permet de définir le numéro d'ID du contenu de la balise, écrit au format *Char_ID* standard. Cet attribut ne doit pas être omis.

[Éléments-enfants]

Le contenu de cet élément peut être toute combinaison de *chaînes de caractères étendus*, d'éléments en-ligne (à l'exception de <page_break>) et de la balise <object>.

<head> </head> Permet de définir une liste de mots clés à utiliser pour effectuer des recherches. Les éléments-enfants ci-dessous sont disponibles. Il est possible d'écrire une chaîne immédiatement après la balise </head>. L'interprétation de ce type de description est laissée au dispositif de visualisation et, le cas échéant à la conversion.

[Éléments-enfants]

<headword> Permet d'enregistrer un seul mot clé. Il doit exister au moins un <headword> par <head>. Ses attributs et éléments-enfants sont les suivants. Il peut également être utilisé pour enregistrer le mot à introduire dans une table de recherche en spécifiant l'attribut table_id.

[Attributs]

type: Permet d'indiquer le type du mot dans une *chaîne de caractères standard*. Les valeurs suivantes sont prédéfinies alors que d'autres valeurs définies par l'utilisateur sont également admises. Toutefois, il est vivement recommandé d'utiliser "pronunciation" lorsque cela est applicable (c'est-à-dire que la chaîne représente des symboles phonétiques).

"pronunciation": La chaîne représente les symboles de la prononciation.

"other values": D'autres valeurs définies par l'utilisateur sont possibles.

table_id: Permet de pointer vers la table de recherche dans laquelle doit être ajouté ce mot clé, à l'aide du numéro d'ID de la table, défini dans <seach_table_def> (voir 10.3.2.6). Ne doit pas être omis si le contenu comporte un objet page de recherche (voir 10.6.7). Ne pas confondre avec Reference ID (voir 10.6.3.4).

phonetic_notation: Permet d'indiquer l'alphabet phonétique utilisé. La valeur par défaut est "IPA". À ignorer si le type n'est pas "pronunciation". Les valeurs suivantes sont prédéfinies alors que d'autres valeurs définies par l'utilisateur sont également admises. Toutefois, il est vivement recommandé d'utiliser "IPA" lorsque cela est raisonnablement applicable.

"IPA": Permet d'indiquer "l'alphabet phonétique international".

[Éléments-enfants]

Cette balise contient le texte qu'il convient d'afficher dans les résultats de la recherche lorsque le mot clé correspond. Il s'agit simplement d'une *chaîne de caractères externes étendus*.

<key> </key> Permet de définir le mot clé actuel. Seuls les caractères admis dans la table de recherche correspondante peuvent être utilisés si **<search_table>** existe. Voir **<enable_key_type>** pour plus de détails en 10.3.2.6. Peut être omis.

[Attribut]

type: Permet d'indiquer le type de mot clé dans une *chaîne de caractères standard*. Peut être omis.

Exemple:

```
<head>
  <headword table_id="ST0001">color</headword>
    <key>color</key>
    <key>colour</key>
  <headword
table_id="ST0002">color,colour</headword>
    <key>カラー</key>
</head>
<meaning> Visual effect caused by the wavelength of light...
```

<meaning> </meaning> Permet de donner des définitions.

[Attributs]

type: Permet d'indiquer la catégorie des données. Il n'existe pas de catégorie prédéfinie.

subid: Permet d'indiquer un Reference ID pour faire référence à une position dans la définition.

level: Permet d'indiquer par un nombre [0-9] le niveau hiérarchique de la signification. Enregistré dans l'ordre d'importance (le plus petit est le plus proche du sommet de la hiérarchie), même s'il n'est pas obligatoire d'utiliser des valeurs consécutives.

no: Permet d'indiquer le nombre de la signification dans le niveau associé désigné par un nombre [0-9].

<example> </example> Permet de donner des exemples. Si un exemple est donné en format bilingue, il peut être représenté ou à l'aide des balises ****, avec l'attribut **lang** mis à la valeur correspondant à la langue, ou encore en format texte avec un délimiteur implicitement prédéfini, selon convenance (voir l'exemple qui suit).

Exemple:

```
<example><span lang="eng" lang_code="ISO 639-3">I'll take
eel.</span><span lang="jpn" lang_code="ISO 639-3">僕はうなぎだ。
</span></example>
<example>I'll take eel|僕はうなぎだ。 </example>
```

<subhead> </subhead> Permet de définir des sous-mots-clefs et les données y associées. Destiné à être utilisé pour introduire des mots dérivés et composés, etc.

[Attributs]

subid: Permet de définir un Reference ID pour cet article. Peut être omis.

type: Permet de définir le type de mot pour le sous-mot-clef (par exemple mot dérivé, mot composé,) dans une *chaîne de caractères standard*. Peut être omis.

[Éléments-enfants]

<subheadword> </subheadword> Permet de définir des sous-mots-clefs (ou sous-mot en tête d'un article de dictionnaire).

[Attributs]

subid: Permet de définir un ID de référence pour le sous-mot-clef. Peut être omis.

type: Permet de définir le type de sous-mot-clef par exemple un mot dérivé, un mot composé, dans une chaîne de caractères standard. Les valeurs suivantes sont prédéfinies, alors que d'autres valeurs définies par l'utilisateur sont également admises. Toutefois, il est vivement recommandé d'utiliser "pronunciation", s'il y a lieu. Peut être omis.

"pronunciation": La chaîne représente les symboles de prononciation.

[Éléments-enfants]

Cette balise contient le texte qu'il convient d'afficher dans les résultats de la recherche lorsque le mot clé correspond. Il s'agit simplement d'une *chaîne de caractères externes étendus*.

<key> </key>

Permet de définir le mot clé actuel. Seuls les caractères admis dans la table de recherche correspondante peuvent être utilisés si **<search_table>** existe. Voir **<enable_key_type>** pour plus de détails en 10.3.2.6. Peut être omis.

[Attribut]

type: Permet d'indiquer le type de mot clé dans une *chaîne de caractères standard*. Peut être omis.

<meaning> Voir l'explication de la balise **<meaning>** ci-dessus pour plus de détails.

<example> Permet de donner des définitions. Voir l'explication de la balise **<example>** ci-dessus pour plus de détails.

<split> </split> Permet de spécifier le délimiteur entre les blocs lexicographiques (cette balise peut être utilisée entre le dernier mot commençant par "A" et le premier mot commençant par "B"). Cette balise permet d'enregistrer les points de départ de ces blocs, et de définir également les chaînes qu'il

convient de rendre à ce titre (par exemple "-B-" pour un bloc composé de mots commençant par "B").

[Attribut]

level: Permet de définir les niveaux hiérarchiques sous forme d'un nombre [0-9]. Enregistré dans l'ordre hiérarchique, c'est-à-dire le niveau de plus petit nombre est le plus proche du sommet de la hiérarchie, même s'il n'est pas obligatoire d'utiliser des valeurs consécutives.

[Éléments-enfants]

Comme pour <p>.

Exemple

```
<dict_body>
<split>- A -</split>
<dic-item> ... </dic-item>
<dic-item> ... </dic-item>
:
<dic-item>...</dic-item>
<split>- B -</split>
<dic-item>...</dic-item>
:
<dict_body>
```

<column> </column> Présente des articles.

[Attributs]

subid: Permet de définir un Reference ID pour la colonne.

type: Permet de définir le type de la colonne dans une *chaîne de caractères standard*.

[Éléments-enfants]

Comme pour <p>.

<page_break/> Permet d'insérer un saut de page. Cette balise comporte les attributs ci-dessous. Elle ne doit pas être utilisée comme élément-enfant de l'une des balises pouvant être utilisées à l'intérieur de <text_body>. Noter que si <page_break/> est le dernier élément de <text_body>, une page blanche est insérée.

[Attribut]

turning_page_control: Permet de définir l'interaction entre le déroulement ordinaire et les sauts de page, en autorisant ou interdisant de traverser le saut de page vers l'avant ou l'arrière. Les possibilités sont indiquées ci-dessous.

"on": Le saut de page ne doit pas être traversé par le déroulement.

"off": Le saut de page peut être traversé par le déroulement.

"forward": Le saut de page peut être traversé par le déroulement uniquement vers l'avant.

"back": Le saut de page peut être traversé par le déroulement uniquement vers l'arrière.

S'il est omis, il convient que le comportement soit conforme aux paramètres globaux définis pour l'*instanciation d'objet texte* (l'attribut *turning_page_control* de *<flow_data>*, voir 10.3.2.4).

Exemple:

```
<page_break/>
<page_break turning_page_control="off"/>
```

c) Objet

<object>...</object> Permet d'insérer un objet externe dans le corps du texte. Ses attributs sont les suivants.

[Attributs]

type: Permet de spécifier le type de l'objet inséré, en donnant le type MIME du fichier. Cet attribut ne doit pas être omis. Dans la version actuelle de la présente norme, les types suivants sont admis:

Image Png: "image/png"

Image Jpeg: "image/jpeg"

Données audio Mp3: "audio/mp3"

Objet d'animation: "application/x-bvf-flip-animation"
(voir 10.6.6)

src: Permet de définir le nom de fichier de l'objet à insérer, écrit au format *Filename* standard. Cet attribut ne doit pas être omis. Il doit être comparé pour vérification à l'attribut de type avant de lire le fichier auquel cet attribut *src* fait référence.

char_id: Permet d'attribuer un numéro d'ID de chaîne à cet objet, écrit comme un *Char_ID*. Peut être omis.

align: Permet de définir la manière dont il convient que le texte encadre l'objet inséré. Les valeurs possibles sont indiquées ci-dessous. S'il est omis, la valeur par défaut est *bottom* pour un texte écrit horizontalement et *middle* pour un texte écrit verticalement.

"top": Affichage en ligne. Si le texte est rédigé horizontalement, il convient d'aligner la partie supérieure de la ligne sur le bord supérieur de l'objet. Si le texte est rédigé verticalement, il convient d'aligner la partie la plus à droite de la colonne sur le bord droit de l'objet.

ABCDEFGH  IJKLMNOP

"middle": Affichage en ligne. Si le texte est rédigé horizontalement, il convient de centrer l'objet verticalement sur la ligne de base de la ligne. Si le texte est rédigé verticalement, il convient de centrer l'objet horizontalement sur la ligne de base de la colonne.



"bottom": Affichage en ligne. Si le texte est rédigé horizontalement, il convient d'aligner la ligne de base de la ligne sur le bord inférieur de l'objet. Si le texte est rédigé verticalement, il convient d'aligner la ligne de base de la colonne sur le bord gauche de l'objet.



"left": Encadrement. Si le texte s'affiche horizontalement, l'objet s'affiche à proximité de la marge de gauche, et le texte est écrit dans l'espace entre l'objet et la marge de droite. Si le texte est écrit verticalement, l'objet s'affiche au dessus, et le texte en-dessous.

Si l'objet est plus large que l'espace d'une ligne/colonne, plusieurs lignes/colonnes de texte peuvent être placées à côté. L'attribut align du paragraphe environnant (balise <p>) n'influence pas la position de l'objet. Même si une autre valeur que "left" lui est attribuée, l'objet s'affiche le long de la marge de gauche. Toutefois, les paramètres du paragraphe s'appliquent toujours au texte entourant l'objet. Si la balise <object> apparaît au milieu d'une ligne de texte, elle s'affiche au début de la ligne suivante.

Il convient que l'exemple ci-dessous soit rendu de la même manière, avec la balise <object> placée à n'importe quelle position entre la lettre "A" et la lettre "M".



"right": Encadrement. Si on affiche un texte horizontal, l'objet est présenté à

proximité de la marge droite, et le texte est écrit dans l'espace entre l'objet et la marge gauche. Si le texte est rédigé verticalement, l'objet est présenté en bas, et le texte au-dessus. Si l'objet est plus grand qu'une ligne/colonne, plus d'une ligne/colonne de texte peuvent être placées à côté.

L'attribut align du paragraphe environnant (balise <p>) n'influence pas la position de l'objet. Même si une autre valeur que "right" lui est attribuée, l'objet s'affiche le long de la marge droite. Toutefois, les paramètres du paragraphe s'appliquent toujours au texte entourant l'objet. Si la balise <object> apparaît au milieu d'une ligne de texte, elle s'affiche sur la droite de la ligne suivante.

Il convient que l'exemple ci-dessous ait la même apparence où que soit la balise <obj> entre les lettres "A" et "M".

ABCDEFGHIJKL	
MNOPQ	
RSTUV	
WXYZ	

start: Si l'attribut src pointe vers un objet animation, il indique quand la lecture démarre. Ne peut être utilisé qu'avec des objets animation. Les valeurs possibles sont indiquées ci-dessous. S'il est omis, la valeur par défaut est "auto". De plus, si la valeur auto lui est attribuée, l'animation ne doit pas être utilisée par la balise <action_play> (voir 10.4.4.2).

"auto": La lecture commence automatiquement, dès que l'objet s'affiche; l'animation est exécutée à partir de son début. Si l'animation est masquée puis à nouveau affichée (à cause d'un texte déroulant), il convient de la faire démarrer de nouveau depuis le début, où qu'elle ait été arrêtée.

"event": La lecture est gérée en fonction des données d'événement (voir 10.4.2).

loop: Permet de définir le nombre de fois qu'il convient de lancer une animation. Dans la présente version de ce document, la seule valeur acceptée est "1". S'il est omis, l'objet d'animation est lu en boucle infinie, jusqu'à ce qu'il sorte de la zone d'affichage. Toutefois, si la valeur "auto" n'est pas affectée à l'attribut start, la valeur de cet attribut loop est ignorée, et le comportement sera identique à ce qu'il serait si la valeur "1" était attribuée.

[Éléments-enfants]

<permission_info> Permet de définir les autorisations relatives à l'objet auquel fait référence l'attribut src. Écrit au format *Permission* standard. Peut être omis.

Exemple:

```
<object type="image/png" src="image1.png"/>
<object type="image/jpeg" src="image2.jpg">
  <permission_info>
    <print_permission permission="authorized"/>
  </permission_info>
</object>
<object type="application/x-bvf-flip-animation"
  src="anime.xml" loop="1"/>
```

d) Éléments du bloc logique

`<div>...</div>` Permet de représenter un bloc logique. Peut être imbriqué.

[Attributs]

level: Permet d'enregistrer par un nombre [0-9] la profondeur du niveau]. Peut être omis. Enregistré par ordre d'importance (plus le nombre est petit plus le bloc est proche du sommet de la hiérarchie), même s'il n'est pas obligatoire d'utiliser des valeurs consécutives.

title: Permet d'enregistrer le titre du bloc. Peut être omis. Il est conçu pour enregistrer cette information.

[Éléments-enfants]

Comme `<p>`.

`<span>...</span>` Balise à utiliser pour distinguer des sous-chaînes qui ne doivent pas apparaître dans le rendu final (les notes de l'éditeur, par exemple). Elle doit être ignorée par le dispositif de visualisation. Elle peut aussi servir de délimiteur dans les balises `<example>`.

[Attributs]

type: Permet de préciser le type de la sous-chaîne. Dans la chaîne de caractères standard. Peut être omis.

lang: Permet de préciser la langue de la sous-chaîne. Peut être omis.

lang_code Permet de préciser le système de codage de langue utilisé. Dans la présente spécification, seul "ISO 639-3" est prédéfini. La valeur par défaut est "ISO 639-3". Peut être omis.

[Éléments-enfants]

Comme pour `<p>`.

Exemple:

```
<example><span lang="eng" lang_code="ISO 639-3">I'll take eel.</span><span
lang="jpn" lang_code="ISO 639-3">僕はうなぎだ。</span></example>
```

e) Éléments table

`<table>...</table>` Permet d'annoncer une table de genre HTML. Cette table vise, à placer, au moment du rendu les données dans des lignes et des colonnes. Ne pas la confondre avec les tables de recherche (voir 10.3.2.6).

[Éléments-enfants]

`<tr>...</tr>` Permet de définir les lignes de la table. `<table>` doit comporter au moins une instanciation de la balise `<tr>` comme élément-enfant.

[Éléments-enfants]

`<td>...</td>` Permet de définir les cellules de la ligne. `<tr>` doit comporter au moins une instanciation de la balise `<td>` comme élément-enfant.

`<th>...</th>` Permet de définir un en-tête pour la table. Peut être omis.

f) Affichage en fonction de l'environnement

`<select>...</select>` Permet de représenter d'autres chaînes à afficher en fonction de l'environnement.

[Éléments-enfants]

`<select_item>` Permet de définir des chaînes pour chaque environnement.

[Attributs]

default: Permet de spécifier que cette instanciation de la balise donne l'environnement par défaut lorsque la valeur attribuée est "yes." Sinon, il est omis.

type: Permet de préciser le type d'environnement (par exemple le nom du matériel auquel est destiné le contenu) dans une *chaîne de caractères standard*. Cet attribut ne doit pas être omis.

g) Fonction TTS (texte-parole)

`<tts>...</tts>` Spécifie la sous-chaîne à insérer dans la représentation du texte à partir de la parole.

[Attribut]

speaker: Permet de spécifier le type de voix. Les valeurs admises sont:

"male":	homme adulte
"female"	femme adulte
"child"	enfant

h) Événement

`<a>` Permet de spécifier une ancre vers un lien comme le fait l'élément `<a>` correspondant du langage HTML.

[Attribut]

href: Permet de spécifier l'emplacement d'une ressource Web.

10.6.3 Instanciation d'un objet données de dictionnaire

10.6.3.1 Généralités

Les instanciations d'un objet données de dictionnaire sont stockées dans leurs propres fichiers XML, l'élément racine étant <dict_data> (voir ci-dessous).

Comme on peut le voir, l'élément <dict_data> comporte deux éléments-enfants principaux, l'un pour stocker les attributs par défaut des données de dictionnaire (<dict_default_attribute>) et l'autre pour stocker les données elles-mêmes (<dict_body>).

10.6.3.2 Attributs de dictionnaire par défaut <dict_default_attribute>

Les attributs de dictionnaire par défaut stockent des paramètres à utiliser lors de l'affichage des données de dictionnaire. A noter que ces paramètres peuvent ne pas toujours être respectés si le dispositif de visualisation n'est pas en mesure de traiter les valeurs/comportements établis dans ce module ou s'il est configuré pour donner la priorité aux paramètres utilisateur. En outre, comme cela a été expliqué en 10.3.2.3, certains des paramètres présentés ici peuvent également être définis dans l'ensemble du flot de texte (dans l'élément-enfant <flow_default_attribute> de <flow_entry>) voir 10.3.2.4. Ils sont utilisés par défaut si aucune valeur n'a été attribuée aux paramètres locaux (ceux définis ici). S'ils sont tous définis, la priorité est accordée aux paramètres locaux.

La balise <dict_default_attribute> comporte les attributs et éléments-enfants suivants.

[Attributs]

- table border="0">
- | | |
| --- | --- |
| baseline: | Permet de définir l'orientation de la ligne de base (et donc du texte) de chaque flot. Les valeurs possibles sont indiquées ci-dessous. Peut être omis. Si ni cet attribut ni l'option correspondante du <flow_default_attribute> (voir 10.3.2.3) ne sont définis, la valeur par défaut dépend du dispositif de visualisation. |
- | | |
| --- | --- |
| "right": | Le sens d'écriture est horizontal (de gauche à droite). Toutefois, il peut être modifié par l'utilisateur. |
- | | |
| --- | --- |
| "right_only": | Le sens de l'écriture est horizontal de gauche à droite et l'utilisateur ne peut pas modifier ce réglage. Mais si l'appareil de visualisation n'a pas ce réglage, ce paramètre peut ne pas être appliqué. |
- | | |
| --- | --- |
| "down": | Le sens d'écriture est vertical (de haut en bas). Toutefois, il peut être modifié par l'utilisateur. |
- | | |
| --- | --- |
| "down_only": | Le sens d'écriture est vertical (de haut en bas), et l'utilisateur ne peut pas modifier ce réglage. Mais si l'appareil de visualisation n'a pas ce réglage, ce paramètre peut ne pas être appliqué.. |
- | | |
| --- | --- |
| valign: | Permet de déterminer comment placer la zone de texte. Ici sont déterminées la position verticale pour un texte écrit horizontalement ou la position horizontale pour un texte écrit verticalement. La valeur décrite ci-dessous remplit les conditions requises. Si elle est omise, il convient d'afficher la zone de texte à partir du haut de la zone d'affichage. |
- | | |
| --- | --- |
| "middle": | Le contenu de la balise <text_body> est centré à l'intérieur de la zone d'affichage. Voir la Figure 5 en 10.6.2.2. Si le |

contenu est plus grand que la zone d’affichage, cet attribut est ignoré.

[Éléments-enfants]

<dict_default_font> Permet de définir la police par défaut et sa couleur pour cet objet données de dictionnaire. S’il est omis, la valeur par défaut est **<flow_default_attribute>**. Ses attributs sont les suivants.

[Attributs]

fontname: Nom de police par défaut. Plusieurs polices peuvent être spécifiées. Dans ce cas, il convient de séparer leurs noms par une virgule (0x2c en Unicode). Par exemple:

fontname="Aaa sans serif,Bbb gothic"

Il convient que le dispositif de visualisation utilise la première police répertoriée disponible. S’il est omis, la valeur par défaut est **<flow_default_attribute>**.

color_space, color, opacity:

Permet de définir la couleur de police à utiliser pour les données de contenu. Écrit au format de données *color* standard. S’il est omis, la valeur par défaut est **<flow_default_attribute>**.

ruby_flag: Permet de définir si le code ruby des données de contenu doit être visualisé ou non. S’il est omis, la valeur par défaut est **<flow_default_attribute>**. Le texte à afficher lorsque le code ruby est actif est celui placé dans la balise **<ruby>** des instanciations d’objet données de dictionnaire. Les attributs suivants peuvent être utilisés.

“yes”: Il convient d’afficher le code ruby, mais il peut être désactivé par l’utilisateur.

“yes_only”: Le code ruby doit être affiché et ne peut pas être désactivé par l’utilisateur. Toutefois, cela ne concerne pas les dispositifs de visualisation qui ne sont pas en mesure d’afficher le code ruby.

“no”: Il est recommandé de ne pas afficher le code ruby, mais il peut toujours être activé en fonction des préférences de l’utilisateur.

“no_only”: Le code ruby ne doit pas être affiché et ne peut pas être désactivé par l’utilisateur. Toutefois, cela ne s’applique pas si le dispositif de visualisation est incapable de désactiver l’affichage du code ruby.

<dict_default_background> Permet de définir la couleur d’arrière-plan et l’image d’arrière-plan à utiliser pour cet objet données de dictionnaire. Si ces deux éléments sont définis, l’image est dessinée centrée avec l’écran rempli de la couleur d’arrière-plan. Enfin, le contenu de la balise **<dict_body>** est représenté par dessus. Si l’image d’arrière-plan est trop grande et ne tient pas dans l’écran, elle est réduite en préservant le rapport hauteur/largeur. Il convient qu’en cas d’omission de cette balise le comportement soit celui indiqué en cas d’omission des attributs de cette balise. Les attributs et éléments-enfants ci-dessous sont disponibles.

[Attributs]

color_space, color, opacity:

Permet de définir la couleur à utiliser en arrière-plan. Écrit au format de données *color* standard. S'il est omis, la valeur par défaut est <flow_default_attribute>.

type:

Permet de stocker le type MIME de l'image d'arrière-plan. Peut être omis uniquement si l'attribut src l'est également. Les valeurs possibles sont indiquées ci-dessous.

"image/png"

"image/jpeg"

src:

Permet de définir le nom de fichier de l'image à utiliser en arrière-plan, écrit au format *Filename* standard. Peut être omis. Avant d'ouvrir le fichier indiqué dans cet attribut, on doit le comparer à l'attribut de type

[Éléments-enfants]

<permission_info>:

Permet de définir les droits d'accès de l'image à laquelle fait référence l'attribut src. Écrit au format *Permission* standard. Peut être omis. Si l'attribut src est omis, il convient d'ignorer ces informations de droit d'accès.

<dict_default_background_music>

Permet de définir la musique d'arrière-plan à jouer lors de l'affichage de cet objet données de dictionnaire. Si cette balise est omise, il convient de ne rien jouer. Les attributs et éléments-enfants ci-dessous sont disponibles.

[Attributs]

type:

Permet de stocker le type MIME de la musique d'arrière-plan. Cet attribut ne doit pas être omis. Les valeurs possibles sont indiquées ci-dessous.

"audio/mp3"

src:

Permet de définir le nom de fichier de la musique d'arrière-plan, écrit au format *Filename* standard. Cet attribut ne doit pas être omis. Avant d'ouvrir le fichier indiqué dans cet attribut, on doit le comparer à l'attribut de type.

loop:

Permet d'indiquer s'il convient de jouer de façon répétitive la musique d'arrière-plan (c'est-à-dire la répéter depuis le début chaque fois que la fin est atteinte). Les valeurs possibles sont "yes" et "no". En cas d'omission, la valeur par défaut est no.

[Éléments-enfants]

<permission_info>:

Permet de définir les autorisations pour le fichier son auquel fait référence l'attribut src. Écrit au format *Permission* standard. Peut être omis.

<line_breaking_method> Permet de spécifier l'algorithme à utiliser pour découper le texte en lignes; des langues différentes ont des pratiques différentes, et cet élément autorise une certaine souplesse. S'il est omis, la valeur par défaut est <flow_default_attribute>. Cette balise comporte les attributs et éléments-enfants suivants.

[Attributs]

method: Permet de choisir la méthode de saut de ligne. La valeur par défaut est "none". Les valeurs possibles sont:

"none": aucun traitement particulier. Quand une ligne est remplie de caractères, passer à la ligne suivante

D'autres valeurs sont possibles pour la localisation linguistique. Voir Article A.2.

D'autres attributs peuvent être ajoutés par des méthodes localisées. Voir Article A.2.

[Éléments-enfants]

Aucun, à l'exception de ceux ajoutés par des méthodes localisées. Voir Article A.2.

10.6.3.3 Partie principale des données de dictionnaire <dict_body>

Stocke la partie principale des données de dictionnaire. Son élément-enfant est le suivant.

[Éléments-enfants]

<dic-item> Permet d'enregistrer des informations pour chacun des articles. Il y a autant de balises que de mots définis. Au moins une instanciation de cette balise doit apparaître dans un contenu.

10.6.3.4 Élément d'article (de dictionnaire) <dic-item>

Les attributs et éléments-enfants de <dic-item> sont les suivants.

[Attributs]

type: Permet d'indiquer le type d'article dans une *chaîne de caractères standard*. Peut être omis.

id: Permet de spécifier un identificateur (Reference ID) pour cet article. Reference ID est utilisé pour:

- a) accéder à l'élément à partir d'autres parties du contenu;
- b) identifier l'élément pour l'édition et la mise à jour du contenu.

Cet attribut ne doit pas être omis. Il s'agit d'une chaîne de caractères alphanumériques qui peut être une chaîne vide ("").

rank Permet d'indiquer par un chiffre [0-9] l'importance de l'article, par exemple pour guider un étudiant.

level Permet d'enregistrer la hiérarchie des articles (hiérarchie chapitre-section, par exemple), qui sans cela serait perdue. A spécifier par un chiffre [0-9]. Enregistré par ordre d'importance (le plus petit nombre pour le plus proche du sommet de la hiérarchie), même s'il n'est pas obligatoire d'utiliser des valeurs consécutives.

page_break Permet d'insérer un saut de page. S'il est omis, sa valeur par défaut est "no."

"yes": insérer un saut de page

"no": ne pas insérer de saut de page

turning_page_control Permet de définir l'interaction entre le déroulement ordinaire et les sauts de page, en autorisant ou interdisant la traversée du saut de page vers l'avant ou vers l'arrière. Les possibilités sont indiquées ci-dessous.

"on": Le saut de page ne doit pas être croisé par le déroulement.

"off":	Le saut de page peut être croisé par le déroulement.
"forward":	Le saut de page peut être croisé par le déroulement uniquement vers l'avant.
"back":	Le saut de page peut être croisé par le déroulement uniquement vers l'arrière.
	S'il est omis, il convient que le comportement soit celui défini par les paramètres globaux définis pour l'instanciation d'objet données de dictionnaire (l'attribut <code>turning_page_control</code> de <code><flow_data></code> , voir 10.3.2.4).
revision	Permet de spécifier le numéro de révision correspondant à l'instanciation de cette balise par un numéro supérieur ou égal à 1. Peut être omis. S'il est omis, il est considéré comme n'étant pas une révision, c'est-à-dire qu'il s'agit de la première version du contenu.
delete	Permet de spécifier l'enlèvement de cette balise pour cause de révision. Les valeurs suivantes sont admises. Quand en affectant la valeur "yes" à son attribut "delete" on a supprimé une certaine instanciation de cette balise, il convient d'ajouter une nouvelle instanciation de la balise <code><dic-item></code> lors de sa restauration dans une révision ultérieure. S'il est omis, sa valeur par défaut est "no."
	"yes": supprimé
	"no": non supprimé

[Éléments-enfants]

Les balises suivantes peuvent apparaître dans `<text_body>` de `<dic-item>`.

`<p>`, `<scrolling_text>`, `<br/>`, `<hr/>`, `<font>`, `<i>`, `<u>`, `<em>`, `<u>`, `<oblq>`, `<sc>`, `<mlg>`, `<horizontal>`, `<ruby>`, `<sub>`, `<sup>`, `<external_char>`, `<mask>`, `<char_id>`, `<head>`, `<object>`, `<div>`, `<table>`...

Note:

- `<head>` doit apparaître une seule fois en haut à l'intérieur de `<dic-item>`;
- dans `<head>`, `<headword>` doit apparaître au moins une fois;
- si la balise `<headword>` apparaît plusieurs fois, les fonctions bookmark et wordbook s'appliquent uniquement à la première;
- `<external_char>` ne doit pas apparaître à l'intérieur de `<headword>`.

[Éléments-enfants]

`<dic-item>` comporte les éléments-enfants suivants différents de ceux présentés ci-dessus.

<code><gender></code>	information sur le genre grammatical. <code><psp></code> peut contenir cette information sans utiliser cette balise, selon la stratégie adoptée par l'éditeur.
<code><glab></code>	Permet d'indiquer une étiquette grammaticale (":AmE", "Abbrev.", par exemple).
<code><pronunciation></code>	Permet d'enregistrer une chaîne alphabétique phonétique. Ses attributs sont présentés ci-dessous. Cette balise a pour objet l'enregistrement des chaînes alphabétiques phonétiques à l'extérieur de <code><headword></code> et de <code><subheadword></code> , si ce type d'utilisation n'est pas interdit. Les attributs ci-dessous sont possibles.

[Attributs]

phonetic_notation: Permet d'indiquer l'alphabet phonétique utilisé. La valeur par défaut est "IPA". Les valeurs suivantes sont prédéfinies alors que d'autres valeurs, définies par l'utilisateur, sont aussi permises. Toutefois, il est vivement recommandé d'utiliser "IPA" lorsqu'il est raisonnablement applicable.

- <psp> Permet d'indiquer une partie d'un discours. D'autres marques grammaticales indiquant le genre et le cas peuvent être incluses ici, en fonction de la stratégie adoptée par l'éditeur.
- <inflec> Permet d'indiquer l'accentuation de la forme verbale.
- <slabel> Permet d'indiquer la catégorie des termes (par exemple "philosophie", "commerce", et leurs abréviations)
- <guideword> Permet d'indiquer également la catégorie des termes. Toutefois, ils sont différents de <slabel>: <slabel> est obligatoire lorsque le (sous) en-tête entre dans l'une des catégories applicables, tandis <guideword> est facultatif et sert à distinguer plusieurs définitions.
- <spellout> Permet de spécifier la chaîne répétée qui doit ou être remplacée par d'autres chaînes (éventuellement un symbole) indiquant une omission, ou simplement répétée.

[Attributs]

org Permet de spécifier la chaîne indiquant une omission. Cette balise vise à préciser la façon de représenter une omission dans le produit final.

Exemple:

```
<example><spellout org = " ~ " >spell</spellout> one's name full
</example>
```



~ one's name full

- <variant> Permet d'indiquer des variantes dans une *chaîne de caractères standard*.
- <etymology> Permet d'indiquer une information étymologique dans une *chaîne de caractères standard*.