

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Multimedia systems and equipment – Multimedia e-publishing and e-books –
Generic format for E-publishing**

**Systèmes et équipements multimédia – Publication et livres électroniques
multimédia – Format générique pour la publication électronique**

IECNORM.COM: Click to view the full PDF of IEC 62448:2009



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2009 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 It also gives information on projects, withdrawn and replaced publications.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 –
Generic format for E-publishing**

**Systèmes et équipements multimédia – Publication et livres électroniques
multimédia – Format générique pour la publication électronique**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

PRICE CODE
CODE PRIX

XH

ICS 33.160.60, 35.240.30

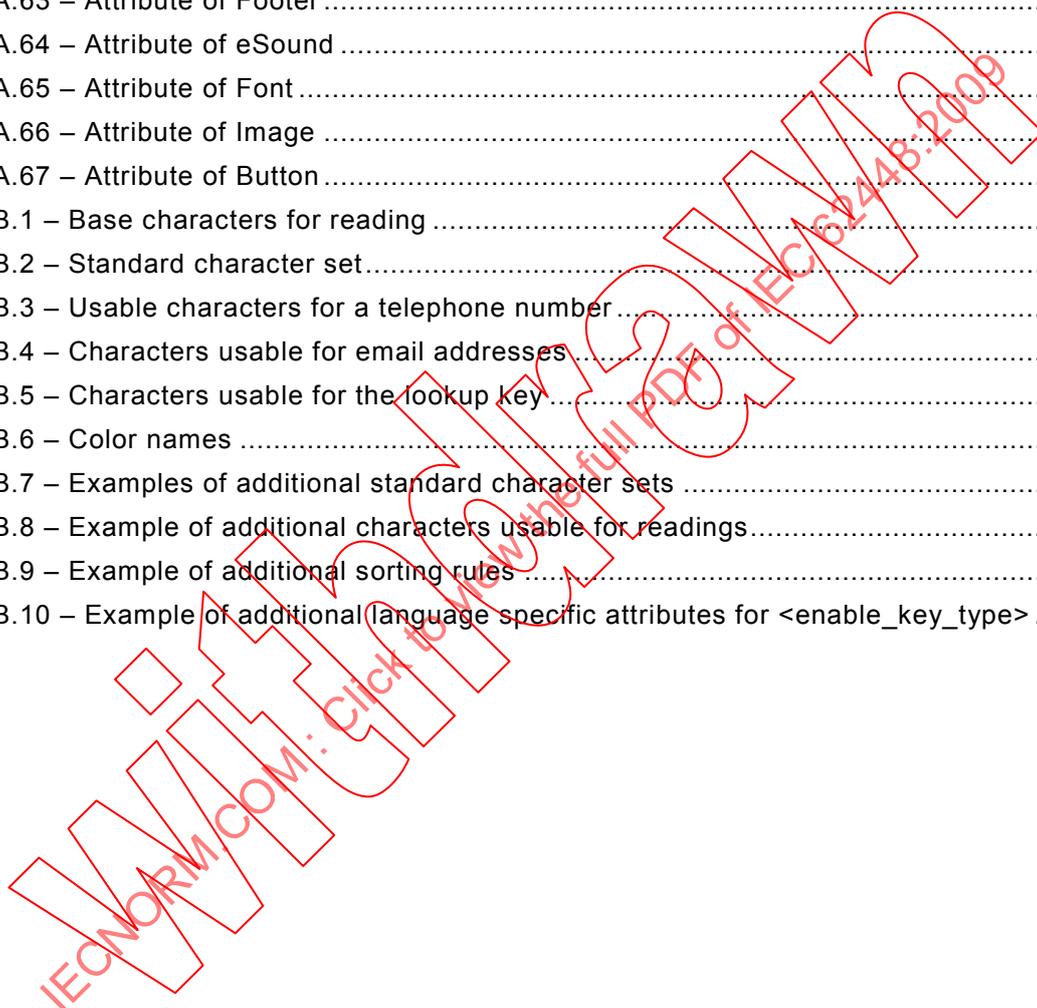
ISBN 978-2-88910-714-8

CONTENTS

FOREWORD.....	5
INTRODUCTION.....	7
1 Scope.....	8
2 Normative references.....	8
3 Terms and definitions.....	8
4 Position and requirements for generic format.....	9
4.1 Generic format in contents creation/distribution model.....	9
4.2 Requirements for generic format.....	9
5 Notation.....	10
6 Logical structure.....	10
7 Semantics.....	11
Annex A (normative) BBeB Xylog Format.....	12
Annex B (normative) XMDF XML format.....	88
Bibliography.....	180
Figure 1 – Contents creation/distribution model.....	9
Figure A.1 – Conceptual diagram of the “Block layout”.....	31
Figure A.2 – Coordinate system.....	32
Figure A.3 – Page layout composition.....	33
Figure A.4 – Block layout composition.....	34
Figure A.5 – Bgimagemode attribute.....	40
Figure A.6 – Layout attribute.....	41
Figure B.1 – XML tree structure.....	88
Figure B.2 – Example of valign=“middle”.....	124
Figure B.3 – Example of dropped capital.....	128
Figure B.4 – Left and right margin of a paragraph.....	128
Figure B.5 – Horizontal writing in vertical text.....	131
Figure B.6 – Ruby.....	132
Figure B.7 – Example of search page object instance rendering.....	140
Figure B.8 – Comic object Instance.....	144
Table A.1 – Data types.....	35
Table A.2 – Attribute of BBeBXylog.....	35
Table A.3 – Attribute of Info.....	36
Table A.4 – Attribute of Title.....	36
Table A.5 – Attribute of Autor.....	36
Table A.6 – Attribute of Publisher.....	37
Table A.7 – Attribute of Label.....	37
Table A.8 – Attribute of Cthumbnail.....	37
Table A.9 – Attribute of TocLabel.....	38
Table A.10 – Attribute of Page.....	39
Table A.11 – Attribute of Wait.....	42

Table A.12 – Attribute of TextBlock	43
Table A.13 – rubyalign attribute	45
Table A.14 – rubyadjust attribute	46
Table A.15 – rubyoverhang attribute	46
Table A.16 – empdotsposition attribute	46
Table A.17 – emplineposition attribute	46
Table A.18 – emplinetype attribute	47
Table A.19 – align attribute	47
Table A.20 – blockrule attribute	48
Table A.21 – Attribute of P	49
Table A.22 – Attribute of Plot	50
Table A.23 – Attribute of Fill	51
Table A.24 – Attribute of Space	51
Table A.25 – Attribute of CharButton	51
Table A.26 – Attribute of Gaiji	52
Table A.27 – Attribute of Box	52
Table A.28 – linetype attribute	53
Table A.29 – Attribute of EmpDots	53
Table A.30 – Attribute of Empline	54
Table A.31 – Attribute of DrawChar	54
Table A.32 – Attribute of ImageBlock	55
Table A.33 – Attribute of ButtonBlock	58
Table A.34 – Attribute of BaseButton	60
Table A.35 – Attribute of FocusinButton	60
Table A.36 – Attribute of JumpTo	60
Table A.37 – Attribute of Run	60
Table A.38 – runoption attribute	61
Table A.39 – Attribute of PushButton	61
Table A.40 – Attribute of UpButton	61
Table A.41 – Attribute of Canvas	62
Table A.42 – Attribute of PutObj	63
Table A.43 – Attribute of Moveto	64
Table A.44 – Attribute of Lineto	64
Table A.45 – arrowtype attribute	65
Table A.46 – Attribute of DrawBox	65
Table A.47 – Attribute of DrawEllipse	66
Table A.48 – Attribute of RuledLine	67
Table A.49 – Attribute of BookStyle	67
Table A.50 – Attribute of SetDefault	68
Table A.51 – setwaitprop attribute	69
Table A.52 – Attribute of RegistFont	69
Table A.53 – Attribute of BookSetting	70
Table A.54 – Attribute of TextStyle	71

Table A.55 – Attribute of BlockStyle	74
Table A.56 – Attribute of PageStyle	76
Table A.57 – Attribute of Window	79
Table A.58 – Attribute of PopUpWin	79
Table A.59 – Attribute of Sound	80
Table A.60 – Attribute of SoundStream	80
Table A.61 – Attribute of ImageStream	81
Table A.62 – Attribute of Header	81
Table A.63 – Attribute of Footer	82
Table A.64 – Attribute of eSound	83
Table A.65 – Attribute of Font	83
Table A.66 – Attribute of Image	84
Table A.67 – Attribute of Button	85
Table B.1 – Base characters for reading	90
Table B.2 – Standard character set	90
Table B.3 – Usable characters for a telephone number	98
Table B.4 – Characters usable for email addresses	98
Table B.5 – Characters usable for the lookup key	113
Table B.6 – Color names	150
Table B.7 – Examples of additional standard character sets	151
Table B.8 – Example of additional characters usable for readings	151
Table B.9 – Example of additional sorting rules	152
Table B.10 – Example of additional language specific attributes for <enable_key_type>	153



INTERNATIONAL ELECTROTECHNICAL COMMISSION

**MULTIMEDIA SYSTEMS AND EQUIPMENT –
MULTIMEDIA E-PUBLISHING AND E-BOOKS –
GENERIC FORMAT FOR E-PUBLISHING**

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 provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
- 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 62448 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.

This second edition cancels and replaces the first edition, published in 2007 and constitutes a technical revision.

The main changes with respect to the previous edition are listed below.

- Addition of Annex B – XMDF XML format
- Related changes in Clauses 2 and 6.

This bilingual version, published in 2009-06, corresponds to the English version.

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

CDV	Report on voting
100/1386A/CDV	100/1501/RVC

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

The French version of this standard has not been voted upon.

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 the maintenance result 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.

IECNORM.COM: Click to view the full PDF of IEC 62448:2009

Withdrawn

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, generic format and reader's format. The submission format has to support an interaction between authors and data preparers. The reader's format depends on e-publishing equipment. The generic format has to provide 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

Sony 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:

Sony Corporation
6-7-35 Kitashinagawa, Shinagawa-ku
JP – Tokyo 141-0001

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.

MULTIMEDIA SYSTEMS AND EQUIPMENT – MULTIMEDIA E-PUBLISHING AND E-BOOKS – GENERIC FORMAT FOR E-PUBLISHING

1 Scope

This International Standard specifies a generic format for multimedia e-publishing employed for e-book data interchange among data preparers and publishers, satisfying a number of publishers requirements: revisable, extensible and heterogeneous logical structure.

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*

ISO 639(all parts), *Codes for the representation of names 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/IEC 10646, *Information technology – Universal Multiple-Octet Coded Character Set (UCS)*
Amendment 1 (2005)
Amendment 2 (2006)
Amendment 3 (2008)
Amendment 4 (2008)
Amendment 5 (2008)

ISO/IEC 19757-2:2003, *Information technology – Document Schema Definition Language (DSDL) – Part 2: Regular-grammar based validation – RELAX NG*

Amendment 1 (2006)

3 Terms and definitions

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

3.1

multimedia e-book

multimedia content consisting of texts, graphics, sounds and/or videos

[IEC 62229, 3.2]

3.2**data preparer**

organization or person that prepares an e-book; an editor is an example of a preparer

[IEC 62229, 3.3 modified]

3.3**publisher**

organization or person that issues and distributes an e-book

[IEC 62229, 3.4]

3.4**generic format**

format for multimedia e-book contents created and edited by a data preparer and modified by a publisher to a reader's format for e-book distribution

4 Position and requirements for generic format**4.1 Generic format in contents creation/distribution model**

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

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

IEC 2379/06

Figure 1 – Contents creation/distribution model

Between the adjacent processing steps, data preparer and publisher, e-book contents data are interchanged using the generic format. Merging original texts, graphics, sounds and/or videos provided by authors, the data preparer creates and edits the e-book data in a generic format. The e-book data are stored and forwarded to the publishers.

Publishers modify the generic format into reader's formats appropriate for distribution schemes and devices. The generic format can be used for e-book distribution as well.

4.2 Requirements for generic format

The generic format can satisfy the following requirements of the data preparer and the publisher:

a) logical structure:

the generic format has to have a logical structure that can easily be revised;

b) style specification:

the logical elements in the generic format should be rendered in accordance with a style specification for creating an appropriate reader's format;

c) heterogeneous structures:

multimedia e-books may include heterogeneous structures such as mathematics, chemistry and music notes according to the genre of contents. Those specific structures should be imported to the generic format;

d) metadata:

the generic format should support some metadata to manage the e-book contents data.

5 Notation

The logical structure of generic format is described by using RELAX NG schema, ISO/IEC 19757-2.

6 Logical structure

The logical structure of generic format: ebook-g, provides a simple and extensible format ebook-g-core and existing and actually employed formats bbebxylg and xmdf. The structures of bbebxylg and xmdf are shown in Annex A and Annex B, respectively.

default namespace= <http://tc100.iec.ch/2005/ebook/generic>

start = ebook-g

ebook-g = ebook-g-core

 | external "bbebxylg.rnc"

 | external "xmdf.rnc"

ebook-g-core = meta-g & body-g

meta-g = external "meta-g.rnc"

body-g = element body-g { body }

body =

 title*,

 foreword*,

 introduction*,

 titled-clause+,

 annex*,

 bibliography*

title = element title {

 element main { text },

 element sub { text }?

 }

foreword = element foreword { text }

introduction = element introduction { block+ }

titled-clause = element clause { id, title, clause-content }

clause-content =

 (titled-clause, titled-clause+) | untitled-clause-content

untitled-clause-content = (untitled-clause, untitled-clause+) | block+

untitled-clause = element clause { id, untitled-clause-content }

referenced-document =

 element referenced-document {

 id,

 element abbrev { text },

 element title { text },

 element field { text }*,

 element url { xsd:anyURI }

 }

annex =

 element annex {

```

    id,
    title,
    clause-content
  }
bibliography = element bibliography { referenced-document+ }
block = p | ol | ul | example | note | pre | float
p = element p { inline }
float = table | figure
table = external "table.rnc"
figure = external "figure.rnc"
ol =
  element ol {
    element li { id, block+ }+
  }
ul =
  element ul {
    element li { block+ }+
  }
example = element example { p+ }
note = element note { p+ }
pre = element pre { pre-content }
pre-content =
  (text
  | element var { pre-content })*
inline =
  (text
  | element code | b | i | var { inline }
  | ref
  | strong)*
ref =
  element ref {
    attribute to { xsd:IDREF }
  }
id = attribute id { xsd:ID }?
strong = element strong { text }

```

7 Semantics

Elements of ebook-g-core should be rendered in accordance with appropriate style specifications. Actual style specifications by style languages, for example XSL or DSSSL, are outside the scope of this standard.

The contents of meta-g.rnc, table.rnc and figure.rnc are not specified within an architecture of the ebook-g-core.

Annex A (normative)

BBeB Xylog Format

A.1 General

This annex specifies the bbebxlog format mentioned in Clause 6.

This format is intended to be used for two kinds of users, that is, the e-Book contents producer and the related tool developer. Those who produce contents can use this specification as a generic contents data storage format based on the international standard specification. The related tool developer can use it as a reference to decide on the specification for the display equipment and to define the conversion rule between this specification and the other data forms such as any types of XML or HTML.

The feature of this specification specified in this annex includes the following:

- a) "expression of book on paper" concerning two screens layout and modified characters, etc.
- b) "expression of digital" such as sound, page jump and interactive action, etc.

The elements necessary for the above-mentioned expressions and their smooth operations are defined in this specification.

A.2 Structure description

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

namespace sch = "http://www.ascc.net/xml/schematron"

DrawChar =

(element.Plot

| element.CR

| element.Fill

| element.CharButton

| element.Yoko

| element.Tate

| element.Nekase

| element.NoBR

| element.DrawChar

| element.Italic

| element.Bold

| SimpleChar1)*

SimpleChar1 =

```
(element.Rubi
| element.Box
| element.EmpDots
| element.EmpLine
| element.Sub
| element.Sup
| element.Space
| SimpleChar0)*
```

```
SimpleChar0 = (text | element.Gaiji | element.AltString)*
```

```
SimpleChar2 = (element.Plot | SimpleChar0)*
```

```
# BBeB XML Xylog Basic
```

```
element.BBeBXylog =
```

```
element BBeBXylog {
  attlist.version,
  element.BookInformation,
  element.Main,
  element.Style,
  element.Objects
}
```

```
attlist.version &=
```

```
attribute version { text }
```

```
>> a:documentation [
```

```
  ' Constraint: Current "version" attribute value is "1.0" '
```

```
]
```

```
element.BookInformation =
```

```
element BookInformation { element.Info, element.TOC }
```

```
element.Info =
```

```
element Info {
```

```
  attlist.version, element.BookInfo, element.DocInfo, element.Keyword*
```

```
}
```

```
element.BookInfo =
```

```
element BookInfo {
```

```
element.Title,  
element.Author,  
element.BookID,  
element.Publisher,  
element.Label,  
element.Category*,  
element.Classification,  
element.FreeText  
}  
element.Title = element Title { attlist.reading, text }  
attlist.reading &= attribute reading { text }  
element.Author = element Author { attlist.reading, text }  
element.BookID = element BookID { text }  
element.Publisher = element Publisher { attlist.reading, text }  
element.Label = element Label { attlist.reading, text }  
element.Category = element Category { text }  
element.Classification = element Classification { text }  
element.FreeText = element FreeText { text }  
element.DocInfo =  
element DocInfo {  
    element.CThumbnail,  
    element.Language,  
    element.Creator,  
    element.SumPage  
}  
element.CThumbnail = element CThumbnail { attlist.CThumbnail }  
attlist.CThumbnail &= attribute file { text }  
element.Language =  
element Language {  
    xsd:string { length = "2" }  
    >> a.documentation [  
        " Constraint: Set two-letter primary language code which is specified in ISO 639."
```

```

    ]
  }
  element.Creator = element Creator { text }
  element.SumPage = element SumPage { text }
  element.Keyword = element Keyword { text }
  element.TOC = element TOC { element.TocLabel* }
  element.TocLabel = element TocLabel { attlist.jump, text }
  element.Main = element Main { element.Page+ }
  element.Page =
    element Page {
      attlist.Page,
      element.Common.Page
      >> sch:pattern [
        name = "Style Check"
        "\x{a}" ~
        " "
        sch:rule [
          context = "Page"
          "\x{a}" ~
          " "
          sch:assert [
            test = "@pagestyle = //Style/PageStyle/@stylelabel"
            "No PageStyle"
          ]
        ]
        "\x{a}" ~
        " "
      ]
      "\x{a}" ~
      " "
    ]
  }
  element.Common.Page =

```

(element.Common.Objects
| element.BlockSpace
| element.RuledLine
| element.Wait)*

attlist.id &= attribute objid { xsd:string }

attlist.refid &= attribute refobj { xsd:string }

attlist.refstream &= attribute refstream { xsd:string }

attlist.Page &=

attlist.id,
attribute pagestyle { text },
attlist.Common.Page

element.TextBlock =

element TextBlock {
attlist.TextBlock,
((element.P | element.CR)*

>> sch:pattern [

name = "Style Check"

"\x{a}" ~

" "

sch:rule [

context = "TextBlock"

"\x{a}" ~

"

sch:assert [

test = "@blockstyle = //Style/BlockStyle/@stylelabel"

"No BlockStyle"

]

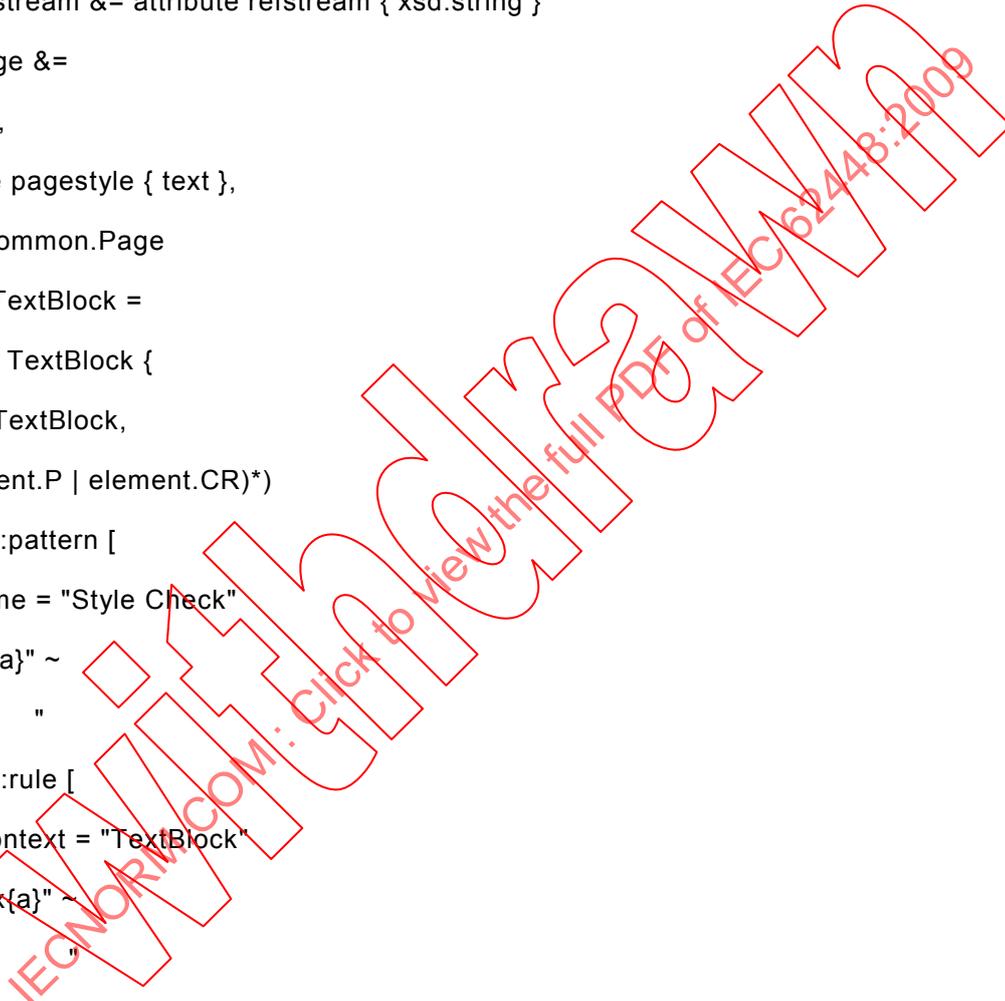
"\x{a}" ~

" "

]

"\x{a}" ~

" "



```

]
>> sch:pattern [
  name = "Style Check"
  "\x{a}" ~
  "    "

  sch:rule [
    context = "TextBlock"
    "\x{a}" ~
    "    "

    sch:assert [
      test = "@textstyle = //Style/TextStyle/@stylelabel"
      "No TextStyle"
    ]
    "\x{a}" ~
    "    "
  ]
  "\x{a}" ~
  "    "
]
}

attlist.TextBlock &=
  attlist.id,
  attribute textstyle { text },
  attribute blockstyle { text },
  attlist.Common.Text,
  attlist.Common.Block

element.P = element P { attlist.P, DrawChar }
attlist.P &= attribute refesound { xsd:string }?
element.Plot = element Plot { attlist.Plot, text }
attlist.xsize &= attribute xsize { xsd:unsignedShort }
attlist.ysize &= attribute ysize { xsd:unsignedShort }
attlist.Plot &=

```

```
attlist.xsize,
attlist.ysize,
attlist.refid,
[ a:defaultValue = "bottom" ]
attribute adjustment { "center" | "baseline" | "top" | "bottom" }?
element.CR = element CR { empty }
element.Fill =
  element Fill {
    attribute code { text },
    empty
  }
element.Space = element Space { attlist.xsize, empty }
element.CharButton = element CharButton { attlist.refid, SimpleChar1 }
element.Rubi = element Rubi { (element.Oyamoji, element.Rubimoji)+ }
element.Oyamoji = element Oyamoji { SimpleChar0 }
element.Gaiji = element Gaiji { attlist.Gaiji, text }
attlist.Gaiji &=
  attlist.refid,
  attribute fontfacename { text }?,
  attribute code { text }
element.AltString = element AltString { element.Org, element.Alt }
element.Org = element Org { text }
element.Alt = element Alt { text }
element.Rubimoji = element Rubimoji { SimpleChar0 }
element.Box = element Box { attlist.Box, SimpleChar0 }
attlist.linetype &=
  attribute linetype { "solid" | "dotted" | "dashed" | "double" }
attlist.Box &= attlist.linetype?
element.EmpDots = element EmpDots { attlist.EmpDots, SimpleChar0 }
attlist.Empdotsposition &=
  attribute empdotsposition { "before" | "after" }
attlist.EmpdotsChar &=
```

```

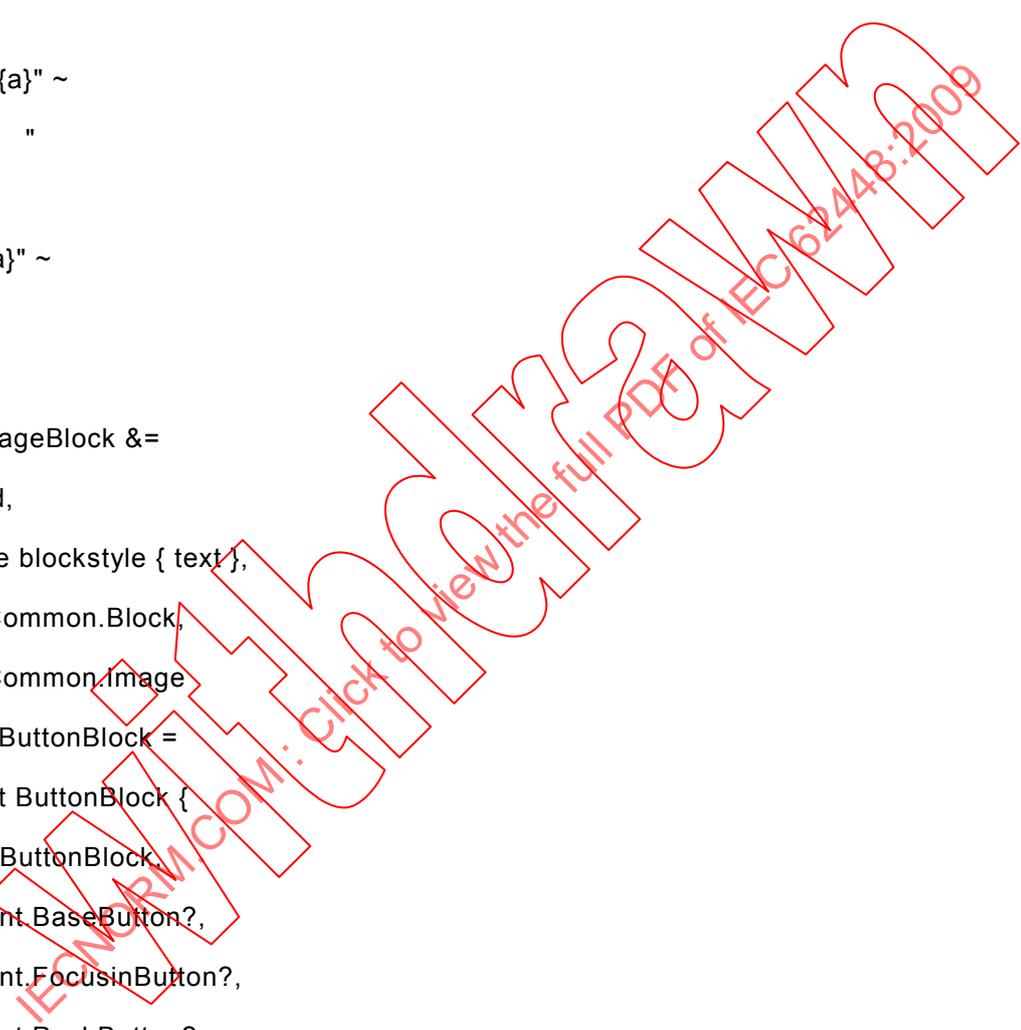
    attribute empdotscode { text },
    attribute empdotsfontname { text },
    attribute refempdotsfont { text }
attlist.EmpDots &= attlist.Empdotsposition?, attlist.EmpdotsChar?
element.EmpLine = element EmpLine { attlist.EmpLine, SimpleChar0 }
attlist.Emplineposition &=
    attribute emplineposition { "before" | "after" }
attlist.Emplinetype &=
    attribute emplinetype {
        "none" | "solid" | "dotted" | "dashed" | "double"
    }
attlist.EmpLine &= attlist.Emplinetype?, attlist.Emplineposition?
element.Sub = element Sub { SimpleChar0 }
element.Sup = element Sup { SimpleChar0 }
element.Yoko = element Yoko { SimpleChar0 }
element.Tate = element Tate { SimpleChar2 }
element.Nekase = element Nekase { SimpleChar2 }
element.NoBR = element NoBR { SimpleChar1 }
element.DrawChar =
    element DrawChar {
        attribute line { xsd:unsignedShort },
        SimpleChar0
    }
element.Italic = element Italic { DrawChar }
element.Bold = element Bold { DrawChar }
element.ImageBlock =
    element ImageBlock { attlist.ImageBlock, text }
>> sch:pattern [
    name = "Style Check"
    "\x{a}" ~
    " "
    sch:rule [

```

```
context = "Page"
"\x{a}" ~
"  "

sch:assert [
  test = "@blockstyle = //Style/BlockStyle/@stylelabel"
  "No BlockStyle"
]
"\x{a}" ~
"  "
]
"\x{a}" ~
"  "
]

attlist.ImageBlock &=
  attlist.id,
  attribute blockstyle { text },
  attlist.Common.Block,
  attlist.Common.Image
element.ButtonBlock =
  element ButtonBlock {
    attlist.ButtonBlock,
    element BaseButton?,
    element.FocusinButton?,
    element.PushButton?,
    (element.UpButton?)
  }
>> sch:pattern [
  name = "Style Check"
  "\x{a}" ~
  "  "
  sch:rule [
    context = "ButtonBlock"
    "\x{a}" ~
```



```

"      "
sch:assert [
  test = "@blockstyle = //Style/BlockStyle/@stylelabel"
  "No BlockStyle"
]
"\x{a}" ~
"      "
]
"\x{a}" ~
"      "
]
}
attlist.ButtonBlock &=
  attlist.id,
  attribute blockstyle { text },
  attlist.Common.Block
attlist.refimage &= attribute refimage { xsd:string }
element.BaseButton = element BaseButton { attlist.refimage?, empty }
element.FocusinButton =
  element FocusinButton {
    attlist.refimage?,
    (element.JumpTo
    | element.Run
    | element.SoundStop
    | element.CloseWindow)*
  }
element.JumpTo = element JumpTo { attlist.jump, empty }
attlist.jump &=
  attribute refpage { xsd:string },
  attlist.refid
element.Run = element Run { attlist.Run, empty }
attlist.Run &=

```

```
[ a:defaultValue = "normal" ]
attribute runoption {
    "normal" | "opposite" | "center" | "opposite-center"
}?,
attlist.refid
element.SoundStop = element SoundStop { empty }
element.CloseWindow = element CloseWindow { empty }
element.PushButton =
    element PushButton {
        attlist.refimage?,
        (element.JumpTo
            | element.Run
            | element.SoundStop
            | element.CloseWindow)*
    }
element.UpButton =
    element UpButton {
        attlist.refimage?,
        (element.JumpTo
            | element.Run
            | element.SoundStop
            | element.CloseWindow)*
    }
element.BlockSpace = element BlockSpace { attlist.BlockSpace, empty }
attlist.BlockSpace &=
    attribute xspace { xsd:short }?,
    attribute yspace { xsd:short }?
element.Canvas =
    element Canvas {
        attlist.Canvas, (element.Common.Canvas | element.Wait)*
    }
element.Common.Canvas =
```

```
element.PutObj
| element.Moveto
| element.Lineto
| element.DrawBox
| element.DrawEllipse
attlist.Canvas &=
  attribute canvaswidth { xsd:unsignedShort },
  attribute canvasheight { xsd:unsignedShort },
  attribute blockrule { "block-fixed" | "block-adjustable" }?,
  attlist.id,
  attlist.Common.Canvas
attlist.Layout &= attribute layout { "LrTb" | "TbRl" }
attlist.Common.Frame &=
  attribute framewidth { xsd:unsignedShort }?,
  attribute framecolor { text }?,
  attribute framemode { "curve" | "square" }?
attlist.Common.Canvas &=
  attribute bgcolor { text }?,
  attlist.Layout?,
  attlist.Common.Frame
element.PutObj = element PutObj { attlist.PutObj, empty }
attlist.x1.unsigned &= attribute x1 { xsd:unsignedShort }
attlist.x1.signed &= attribute x1 { xsd:short }
attlist.y1.unsigned &= attribute y1 { xsd:unsignedShort }
attlist.y1.signed &= attribute y1 { xsd:short }
attlist.PutObj &=
  attlist.x1.unsigned, attlist.y1.unsigned, attlist.refid
element.Wait =
  element Wait {
    attribute time { xsd:short },
    empty
  }
}
```

element.Moveto =

element Moveto { attlist.x1.signed, attlist.y1.signed, empty }

element.Lineto = element Lineto { attlist.Lineto, empty }

attlist.Common.line &=

attribute linewidth { xsd:unsignedShort }?,

attribute linecolor { text }?,

attlist.linetype?

attlist.Lineto &=

attlist.x1.signed,

attlist.y1.signed,

attlist.Common.line,

attribute arrowtype { text }?

element.DrawBox = element DrawBox { attlist.DrawBox, empty }

attlist.DrawBox &=

attlist.x1.signed,

attlist.y1.signed,

attlist.Common.line,

attribute fillcolor { text }?

element.DrawEllipse = element DrawEllipse { attlist.DrawEllipse, empty }

attlist.DrawEllipse &=

attlist.x1.signed,

attlist.y1.signed,

attlist.Common.line,

attribute fillcolor { text }?

element.RuledLine = element RuledLine { attlist.RuledLine, empty }

attlist.RuledLine &=

attlist.linetype,

attribute linewidth { xsd:unsignedShort },

attribute linelength { xsd:unsignedShort },

attribute linecolor { text }?

element.Style =

element Style {

```

    element.BookStyle,
    (element.TextStyle | element.BlockStyle | element.PageStyle)*
}

element.BookStyle =
  element BookStyle {
    attlist.Style,
    element.SetDefault?,
    element.RegistFont*,
    element.BookSetting
  }

attlist.Style &=
  attribute stylelabel { text },
  attlist.id
element.SetDefault = element SetDefault { attlist.SetDefault, empty }
attlist.Rubyoverhang &= attribute rubyoverhang { "auto" | "none" }
attlist.Rubyalign &= attribute rubyalign { "start" | "center" }
attlist.Rubyadjust &= attribute rubyadjust { "line-edge" | "none" }
attlist.Setwaitprop &= attribute setwaitprop { "replay" | "noreplay" }
attlist.SetDefault &=
  attlist.Rubyalign,
  attlist.Rubyoverhang,
  attlist.Empdotsposition,
  attlist.Emplineposition,
  attlist.Emplinetype,
  attlist.EmpdotsChar,
  attlist.Rubyadjust,
  attlist.Setwaitprop
element.RegistFont = element RegistFont { attlist.RegistFont, empty }
attlist.FontEncoding &= attribute encoding { "TTF" | "OTF" | "BF" }
attlist.RegistFont &=
  attribute fontfilename { text },
  attribute file { text },

```

attribute fontname { text },
attlist.FontEncoding
element.BookSetting = element BookSetting { attlist.BookSetting, empty }
attlist.BookSetting &=
attribute bindingdirection { "Lr" | "Rl" },
attribute dpi { xsd:unsignedShort },
attribute screenwidth { xsd:unsignedShort },
attribute screenheight { xsd:unsignedShort },
[a:defaultValue = "24"] attribute colordepth { xsd:unsignedByte }?
element.TextStyle = element TextStyle { attlist.TextStyle, empty }
attlist.TextStyle &= attlist.Style, attlist.Common.Text
attlist.Common.Text &=
attlist.EmpDots,
attlist.EmpLine,
attribute fontsize { xsd:short }?,
attribute fontwidth { xsd:short }?,
attribute fontescapement { xsd:short }?,
attribute fontorientation { xsd:short }?,
attribute fontfacename { text }?,
attribute textcolor { text }?,
attribute textbgcolor { text }?,
attribute wordspace { xsd:short }?,
attribute letterspace { xsd:short }?,
attribute baselineskip { xsd:short }?,
attribute linespace { xsd:short }?,
attribute parindent { xsd:short }?,
attribute parskip { xsd:short }?,
attlist.Rubyalign?,
attlist.Rubyoverhang?,
attribute column { text }?,
attribute columnsep { text }?,
attribute align { "head" | "center" | "foot" }?,

attribute linecolor { text }?,
attribute charspace { xsd:short }?,
attribute fontweight { xsd:unsignedShort }?,
attlist.Rubyadjust?,
attribute textlinewidth { xsd:unsignedShort }?
element.BlockStyle = element BlockStyle { attlist.BlockStyle, empty }
attlist.BlockStyle &= attlist.Style, attlist.Common.Block
attlist.Common.Block &=
attribute blockwidth { xsd:unsignedShort }?,
attribute blockheight { xsd:unsignedShort }?,
attribute blockrule {
"horz-fixed"
| "horz-adjustable"
| "vert-fixed"
| "vert-adjustable"
| "block-fixed"
| "block-adjustable"
}?,
attribute bgcolor { text }?,
attlist.Layout?,
attlist.Common.Frame,
attribute topskip { xsd:unsignedShort }?,
attribute sidemargin { xsd:unsignedShort }?,
attribute footskip { xsd:unsignedShort }?,
attribute refbgimage { text }?,
attlist.Bgimagemode?
element.PageStyle = element PageStyle { attlist.PageStyle, empty }
attlist.PageStyle &=
attlist.Style,
attlist.Common.Page,
attribute setemptyview { "show" | "empty" }?,
attlist.Setwaitprop?

attlist.Bgimagemode &=
attribute bgimagemode { "fix" | "tile" | "centering" }

attlist.Common.Page &=
(attribute refbgimage { text },
attlist.Bgimagemode)?,
attribute evenfooterid { xsd:string }?,
attribute evenheaderid { xsd:string }?,
attribute oddfooterid { xsd:string }?,
attribute oddheaderid { xsd:string }?,
attribute pageposition { "upper" | "lower" | "any" }?,
attribute topmargin { xsd:short }?,
attribute headheight { xsd:short }?,
attribute headsep { xsd:short }?,
attribute oddsidemargin { xsd:unsignedShort }?,
attribute evensidemargin { xsd:unsignedShort }?,
attribute textheight { xsd:unsignedShort }?,
attribute textwidth { xsd:unsignedShort }?,
attribute footheight { xsd:short }?,
attlist.Layout?,
attribute footspace { xsd:short }?

element.Common.Objects &=

element.ImageBlock
| element.TextBlock
| element.ButtonBlock
| element.Canvas

element.Objects =

element Objects {
(element.Common.Objects
| element.Window
| element.PopUpWin
| element.Sound
| element.SoundStream

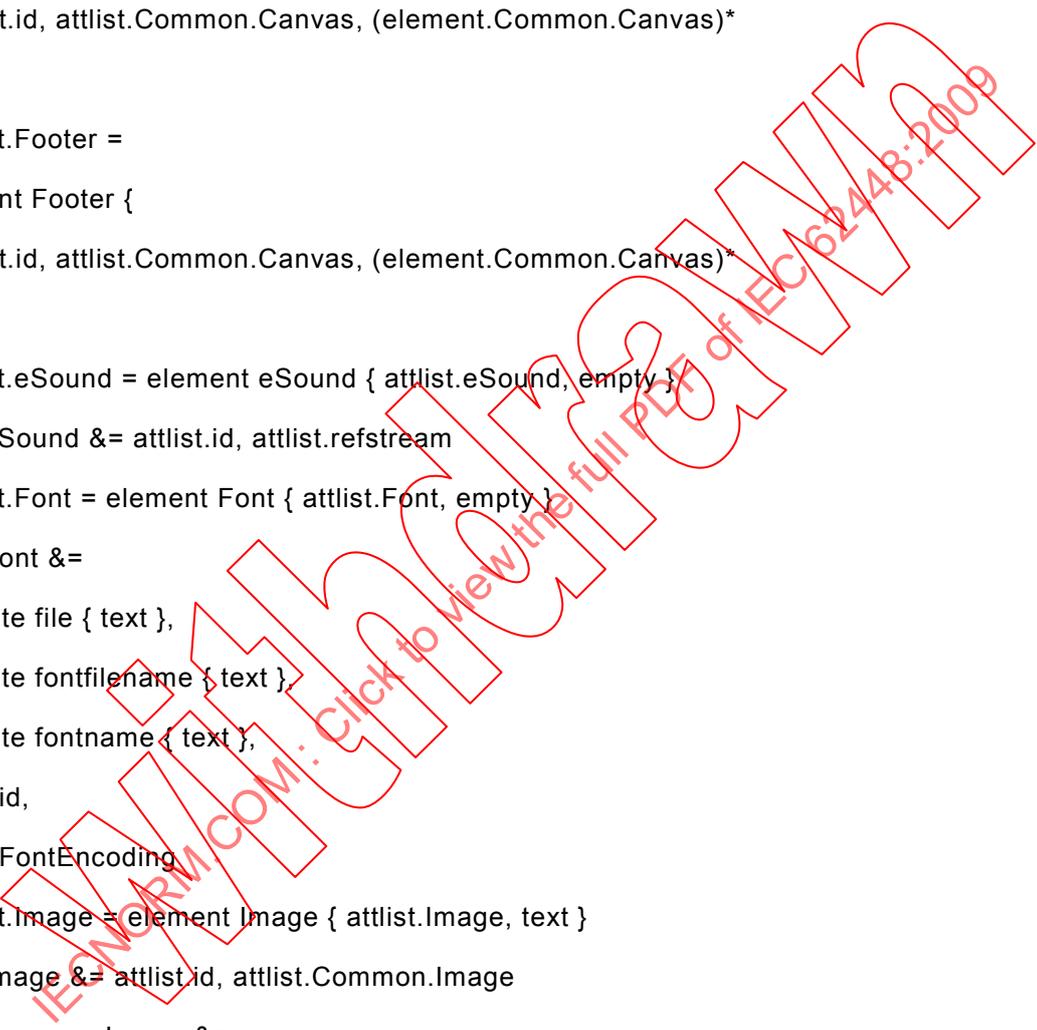
```

    | element.ImageStream
    | element.Header
    | element.Footer
    | element.eSound
    | element.Font

    | element.Image
    | element.Button)*
}
element.Window = element Window { attlist.Window, element.CommonPage }
attlist.Window &=
    attribute windowwidth { xsd:unsignedShort }?,
    attribute windowheight { xsd:unsignedShort }?,
    attlist.Layout?,
    attlist.id
element.PopUpWin =
    element PopUpWin {
        attlist.PopUpWin, (element.TextBlock | element.ImageBlock)
    }
attlist.PopUpWin &= attlist.id
element.Sound = element Sound { attlist.Sound, empty }
attlist.Sound &=
    [ a:defaultValue = " 1" ] attribute times { xsd:unsignedShort }?,
    [ a:defaultValue = "sync" ] attribute playmode { "sync" | "async" }?,
    attlist.refstream,
    attlist.id
element.SoundStream = element SoundStream { attlist.SoundStream, empty }
attlist.SoundStream &=
    attribute encoding { "PCM" | "MP3" },
    attribute file { text },
    attlist.id
element.ImageStream = element ImageStream { attlist.ImageStream, text }

```

```
attlist.ImageStream &=
  attribute encoding { "JPEG" | "GIF" | "BMP" | "PNG" },
  attribute file { text },
  attlist.id
element.Header =
  element Header {
    attlist.id, attlist.Common.Canvas, (element.Common.Canvas)*
  }
element.Footer =
  element Footer {
    attlist.id, attlist.Common.Canvas, (element.Common.Canvas)*
  }
element.eSound = element eSound { attlist.eSound, empty }
attlist.eSound &= attlist.id, attlist.refstream
element.Font = element Font { attlist.Font, empty }
attlist.Font &=
  attribute file { text },
  attribute fontfilename { text },
  attribute fontname { text },
  attlist.id,
  attlist.FontEncoding
element.Image = element Image { attlist.Image, text }
attlist.Image &= attlist.id, attlist.Common.Image
attlist.Common.Image &=
  attribute x0 { xsd:unsignedShort },
  attribute y0 { xsd:unsignedShort },
  attlist.x1.unsigned,
  attlist.y1.unsigned,
  attlist.xsize,
  attlist.ysize,
  attlist.refstream
element.Button =
```



```

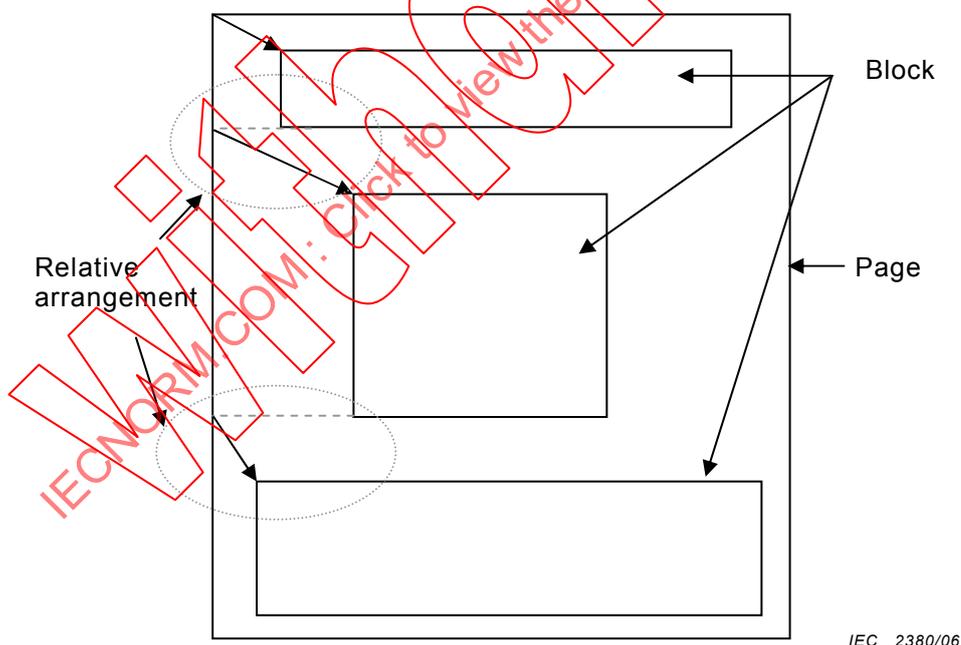
element Button {
  attlist.Button,
  element.BaseButton?,
  element.FocusinButton?,
  element.PushButton?,
  element.UpButton?
}
attlist.Button &= attlist.id
start = element.BBeBXylog

```

A.3 Block layout

A.3.1 Layout

The “BBeB Book Format” utilizes the “Block layout” concept. In the “Block layout”, the content to be shown is displayed in an area called “Block”. The layout is composed of multiple “Block”s. Each “Block” is placed in a relative location. The “BBeB XML Xylog” file format defines the implementation rules based on this “Block layout” concept, as shown in Figure A.1.



IEC 2380/06

Figure A.1 – Conceptual diagram of the “Block layout”

A.3.2 Block

This specification defines the **TextBlock**, **ImageBlock**, **ButtonBlock**, and **Canvas** elements. These are positioned on a “Page” as a “Block”. These are described as sub-elements of the **Page** element with the **BlockSpace** element that indicates the relative arrangement position.

It is possible to specify background image, background color and frame in the “Block”.

A.3.3 Coordinate system

This standard uses a two-dimensional coordinate system (X,Y) that depends on the vertical layout or the horizontal layout, see Figure A.2.

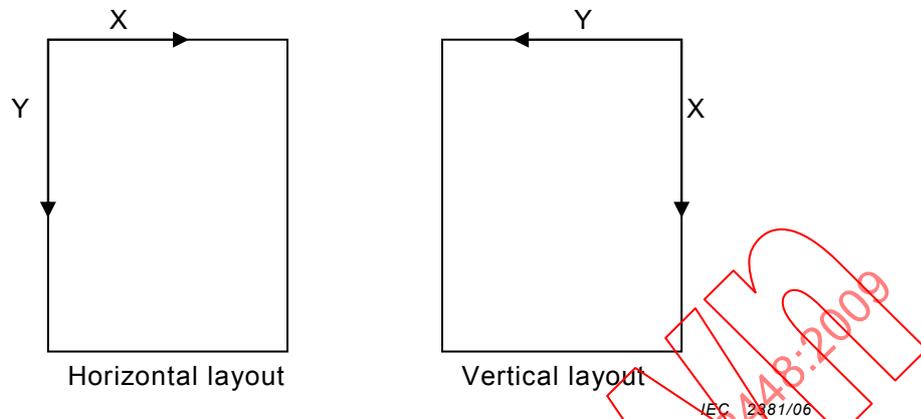


Figure A.2 – Coordinate system

A.4 Terminology

A.4.1 Page layout

For the page layout area, the following terms are defined (see Figure A.3):

Screen size:

screen size of page content which is expected to be viewed by a specific display device and specified by the value of the **screenwidth** and **screenheight** attribute in the **BookSetting** element (see A.5.71)

topmargin:

the space between the top edge of screen and Header area

Header area:

area where header(s) is/are placed

headheight:

the height of Header area

headsep:

the space between Header area and Main text area

Main text area:

area where main content is placed

textheight:

the height of Main text area

textwidth:

the width of Header area, Main text area and Footer area

footspace:

the distance between the bottom of Main text area and the bottom edge of Footer area

footheight:

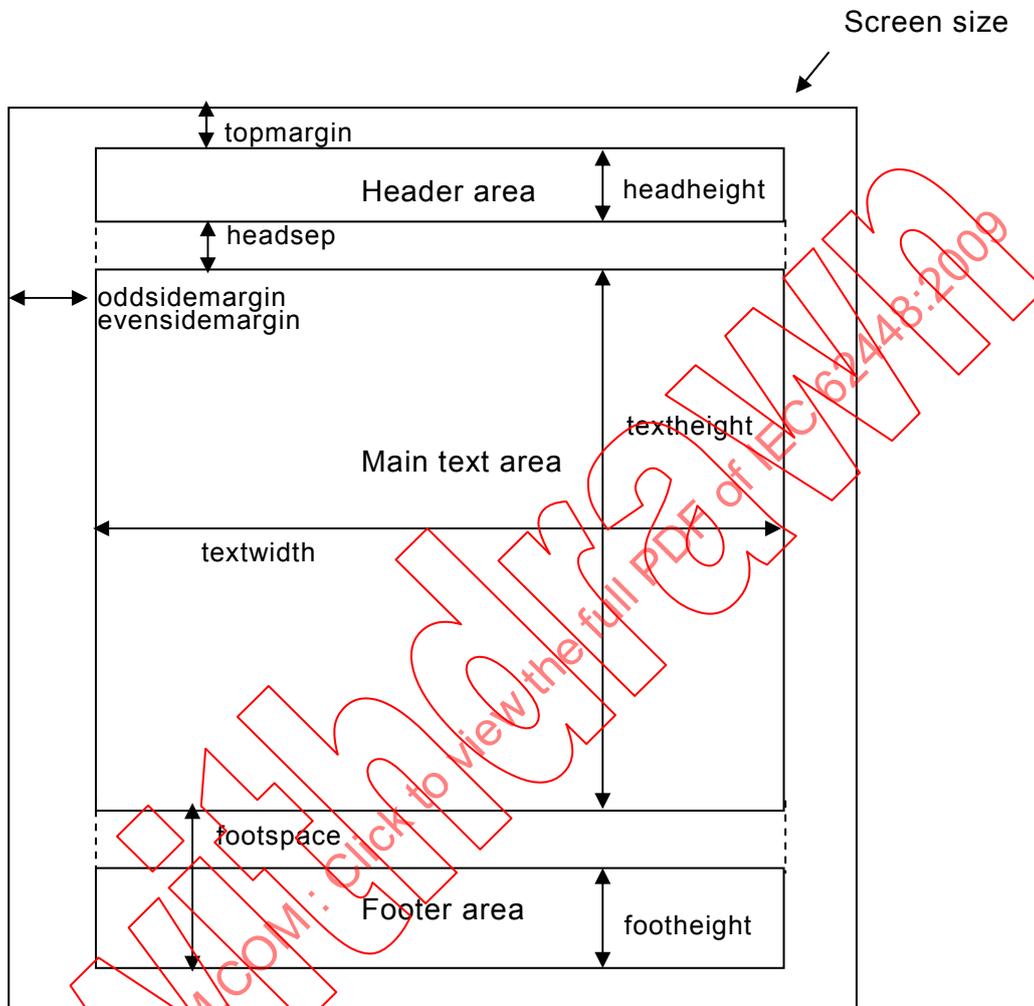
the height of Footer area

oddsidemargin:

the margin between the left-hand side edge of screen and left-hand side edge of Main text area in the odd page

evensidemargin:

the margin between the left-hand side edge of screen and left-hand side edge of Main text area in the even page.



IEC 2382/06

Figure A.3 – Page layout composition

A.4.2 Block layout

For the Block layout area, the following terms are defined (see Figure A.4):

rendering area:

area defined in a block, where the text or image are placed

topskip:

margin between the top edge of the block and the top edge of the rendering area

sidemargin:

margin between the side edge of a block and the side edge of the rendering area for both sides

footskip:

margin between the bottom edge of the block and the bottom edge of the rendering area

blockheight:

height of the “Block”

blockwidth:

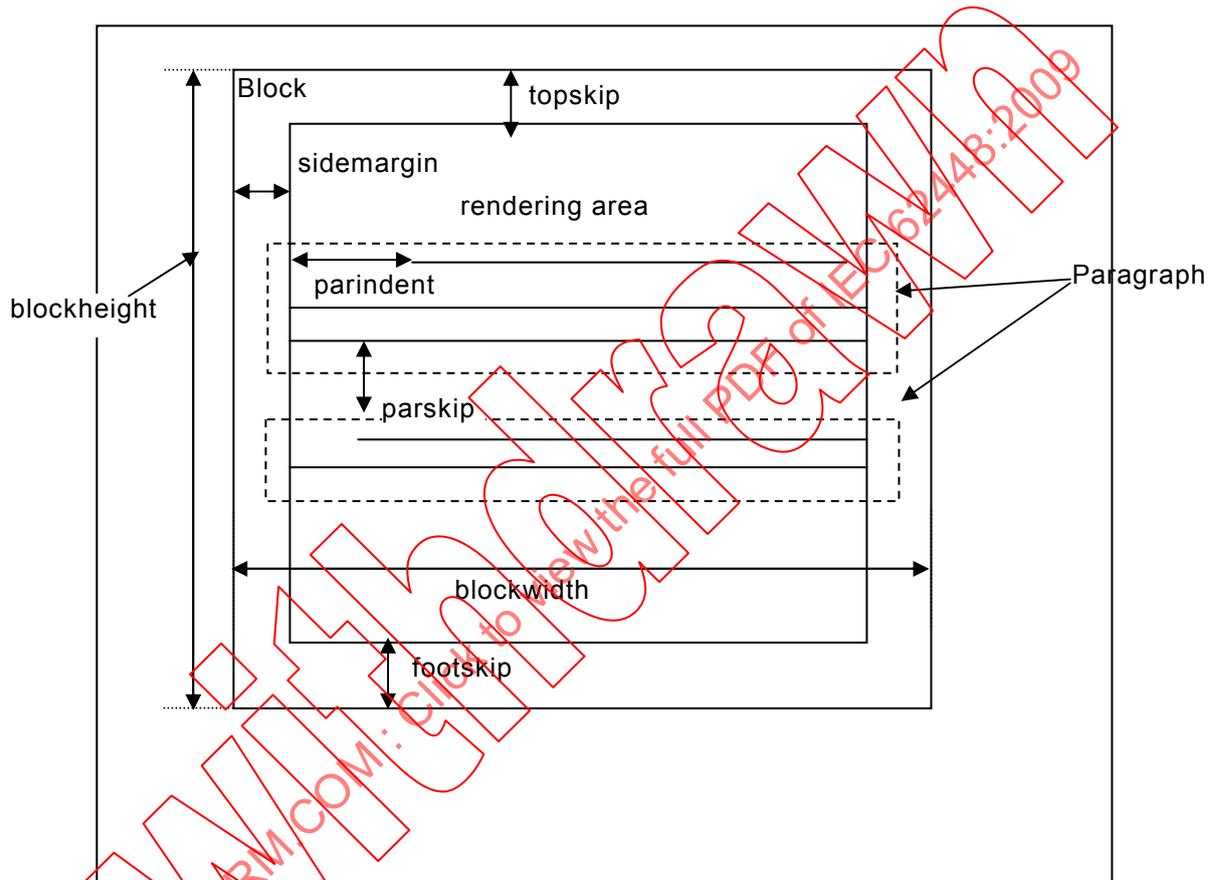
width of the “Block”

parindent:

indent of the paragraph measured from the side edge of the rendering area

parskip:

distance between the end line of a paragraph and the start line of the next paragraph



IEC 2383/06

Figure A.4 – Block layout composition

A.4.3 Data types

The following type definitions are defined as the XML “Attribute” value in this standard.

Table A.1 – Data types

Type definition	Meaning	Example
String	Specifies a normal string of characters	“after”, “solid”
String for sorting	Only able to use the character string defined in the “Character Set for Sorting Specification.	“Kennedy John F.”
File path	Specifies the path in which the file is located	“D:\sample.jpg”, “..\data\white.mp3”
Decimal string	Specifies a decimal numeral as a character string	“10”, “20”
Hexadecimal string	Specifies a hexadecimal numeral as a character string. “0x” is placed at the start of the numeral to show it is hexadecimal	“0x12”, “0x3333”
Style string	Specifies a character string used in style labels	“Main Block”, “Cover Page”, “Section Title Text”
Object ID string	Specifies a character string used in object IDs	“33”, “Object12”
COLORREF string	Specifies 32-bit hexadecimal character string signifying color. The description format is 0xaaarrggbb. (aa: alpha channel [specified as “00” or “ff” in these specifications], “rr”: red value, “gg”: green value, “bb”: blue value)	“0x00ff0000” (red)

A.5 Elements and attributes

A.5.1 BBeBXylog

The **BBeBXylog** element is an element that indicates the area of a whole content in accordance with “BBeB XML Xylog file format”.

Table A.2 – Attribute of BBeBXylog

Attribute name	Default value	Data type	Comments
version (required)		String	Defines as “1.0”.

The **version** attribute specifies the relevant version number for the “BBeB XML Xylog” file format. The value shall be “1.0” by this standard.

A.5.2 BookInformation

The **BookInformation** element specifies bibliographical information and the TOC (table of content) information.

A.5.3 Info

The **Info** element specifies the bibliographical information and the additional information of the content.

Table A.3 – Attribute of Info

Attribute name	Default value	Data type	Comments
version (required)		String	Defines as "1.0".

The **version** attribute specifies the version of the bibliographical information format. The value shall be "1.0" according to this standard.

A.5.4 BookInfo

The **BookInfo** element specifies the bibliographical information of the content.

A.5.5 Title

The **Title** element specifies the title of the content.

Table A.4 – Attribute of Title

Attribute name	Default value	Data type	Comments
reading (required)		String for sorting	

The **reading** attribute specifies the string to sort the content by the title. "Japanese syllabary character", etc. can be used in Japanese. "Alphabet", etc. can be used in English. The detail of the character sets used in this attribute may be specified in interchange parties.

A.5.6 Author

The **Author** element specifies the author name of the content. If the user wants to describe plural author names, the user must describe author names serially in an element.

Table A.5 – Attribute of Autor

Attribute name	Default value	Data type	Comments
reading (required)		String for sorting	

The **reading** attribute specifies the string to sort the content by the author. "Japanese syllabary character", etc. can be used in Japanese. "Alphabet" etc. can be used in English. The detail of the character sets used in this attribute may be specified in interchange parties.

A.5.7 BookID

The **BookID** element specifies ID information of the content. This ID is used to distinguish "Personal content" and the "Commercial content". BookID for the Personal content shall be specified by the character "FB" and the following characters of 14 digits or less (16 digits or less in total). BookID for the Commercial content may be specified in Interchange Parties.

A.5.8 Publisher

The **Publisher** element specifies the publisher name of the content.

Table A.6 – Attribute of Publisher

Attribute name	Default value	Data type	Comments
reading (required)		String for sorting	

The **reading** attribute specifies the string to sort the content by the publisher. "Japanese syllabary character", etc. can be used in Japanese. "Alphabet", etc. can be used in English. The detail of the character sets used in this attribute may be specified in interchange parties.

A.5.9 Label

The **Label** element specifies the label name.

Table A.7 – Attribute of Label

Attribute name	Default value	Data type	Comments
reading (required)		String for sorting	

The **reading** attribute specifies the string to sort the content by the label. "Japanese syllabary character", etc. can be used in Japanese. "Alphabet", etc. can be used in English. The detail of the character sets used in this attribute may be specified in interchange parties.

A.5.10 Category

The **Category** element specifies the genre of the content. The number of this element shall be up to two.

A.5.11 Classification

The **Classification** element specifies information on what kind of data is included in the content (e.g. sound, color image).

A.5.12 FreeText

The **FreeText** element specifies free description about the content (e.g. content summary).

A.5.13 DocInfo

The **DocInfo** element specifies the thumbnail, language information, etc.

A.5.14 Cthumbnail

The **Cthumbnail** element specifies the file name of the thumbnail image for the content.

Table A.8 – Attribute of Cthumbnail

Attribute name	Default value	Data type	Comments
file (required)		File path	Specifies the absolute path or the accessible relative path of the file to be used as a thumbnail image

The **file** attribute specifies the file name and path in which the thumbnail file is located.

A.5.15 Language

The **Language** element specifies the main language used in the content. It should use “ISO 639 language codes”, for example Japanese: “ja”.

A.5.16 Creator

The **Creator** element specifies the creator or studio name of the content.

A.5.17 SumPage

The **SumPage** element specifies the number of “View”s (visible pages for the **BookSetting** element conditions) of the content.

A.5.18 Keyword

The **Keyword** element specifies keywords necessary for searching the content.

A.5.19 TOC

The **TOC** element specifies the table of the content information.

A.5.20 TocLabel

The **TocLabel** element specifies the character string shown as the table of content list.

Table A.9 – Attribute of TocLabel

Attribute name	Default value	Data type	Comments
refobj (required)		Object ID String	Specifies the “objid” of TextBlock , ImageBlock , ButtonBlock and Canvas elements
refpage (required)		Object ID string	Specifies the “objid” of the Page element

A.5.20.1 refobj

The **refobj** attribute specifies the “objid” of the element specified as the “Jump” target. The element specified as “objid” should be located in the **Page** element specified as “refpage”.

A.5.20.2 refpage

The **refpage** attribute specifies the “objid” of the Page element including the element specifying as the “Jump” target.

A.5.21 Main

The **Main** element specifies the “Main content”.

A.5.22 Page

The **Page** element specifies the elements composed of the “Page” and the its layout information itself. The attribute of the **Page** element is recommended to describe the only different information from the **PageStyle** element information specified by “pagestyle”.

Table A.10 – Attribute of Page

Attribute name	Default value	Data type	Comments
bgimagemode		String	Specifies “fix”, “tile” or “centering”
evenfooterid		Object ID string	Specifies the “objid” of the Footer element
evenheaderid		Object ID string	Specifies the “objid” of the Header element
evensidemargin	(Unsigned integer)	Decimal string	[dot]
footheight	(Signed integer)	Decimal string	[dot]
footspace	(Signed integer)	Decimal string	[dot]
headheight	(Signed integer)	Decimal string	[dot]
headsep	(Signed integer)	Decimal string	[dot]
layout		String	Specifies “LrTb” or “TbRI”
Objid (Required)		String	Specifies the only character string in the file
oddfooterid		Object ID string	Specifies the “objid” of the Footer element
oddheaderid		Object ID string	Specifies the “objid” of the Header element
oddsidemargin	(Unsigned integer)	Decimal string	[dot]
pageposition		String	Specifies “upper”, “lower” or “any”
Pagestyle (Required)		Style string	It is necessary to match the specified style character string to “stylelabel” of the referred PageStyle element
refbgimage		Object ID string	Specifies the “objid” of the Image element. Moreover, it is necessary to specify the bgimagemode attribute at the same time
textheight	(Unsigned integer)	Decimal string	[dot]
textwidth	(Unsigned integer)	Decimal string	[dot]
topmargin	(Signed integer)	Decimal string	[dot]

A.5.22.1 pagestyle

The **pagestyle** attribute specifies the “stylelabel” of the **PageStyle** element with necessary information for the “Page” layout.

A.5.22.2 refbgimage

The **refbgimage** attribute specifies the “objid” of the **Image** element to be shown as the background of the **Page** element. When the “objid” is not specified, the image is not displayed.

A.5.22.3 pageposition

The **pageposition** attribute specifies whether the start position is the left page or the right page on the two screen display.

Upper is specified when it is shown on the upper side, lower is specified when it is shown on the lower side, and any is specified when it is shown after the preceding page.

The **bindingdirection** attribute “RI” (mostly vertical flow content) in the **BookSetting** element:

“upper” means the start page is the right.

“lower” means the start page is the left.

“any” means to display continuously from the previous page.

The **bindingdirection** attribute “Lr” (mostly horizontal flow content) in the **BookSetting** element:

“upper” means the start page is the left.

“lower” means the start page is the right.

“any” means to display continuously from the previous page.

A.5.22.4 oddheaderid

The **oddheaderid** attribute specifies the “objid” of the **Header** element shown in the odd “Page” (upper page) on the 2 screen display.

A.5.22.5 evenheaderid

The **evenheaderid** attribute specifies the “objid” of the **Header** element shown in the even “Page” (lower pages) on the 2 screen display.

A.5.22.6 oddfooterid

The **oddfooterid** attribute specifies the “objid” of the **Footer** element shown in the odd “Page” (upper page) on the 2 screen display.

A.5.22.7 evenfooterid

The **evenfooterid** attribute specifies the “objid” of the **Footer** element shown in the even “Page” (lower pages) on the 2 screen display.

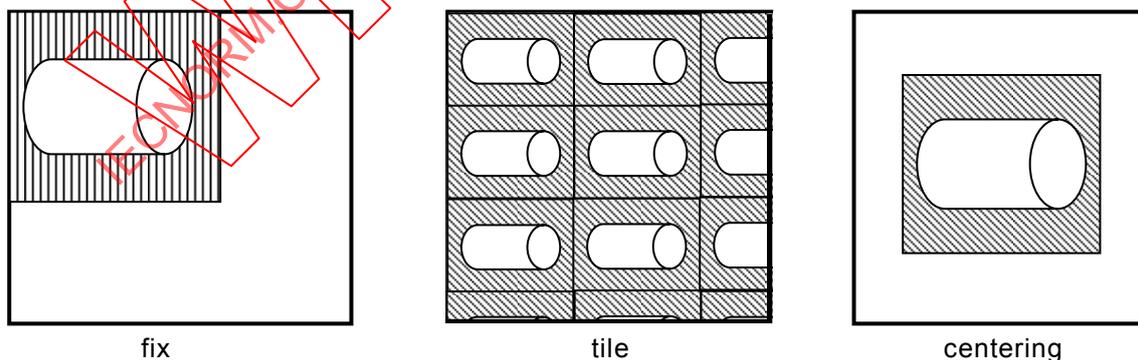
A.5.22.8 bgimagemode

The **bgimagemode** attribute specifies how the “Image” specified in the **refbgimage** attribute should be positioned.

The value is “fix”: The image position is at the origin of the “main text area”.

The value is “tile”: The images fill the “main text area” from the origin.

The value is “center”: The image position is the center of the “main text area”.



IEC 2384/06

Figure A.5 – Bgimagemode attribute

A.5.22.9 evensidemargin

The **evensidemargin** attribute sets the left margin of the “Page layout composition” on the 2 screen display: the left page in the horizontal layout, and the right page in the vertical layout.

A.5.22.10 footnote

The **footnote** attribute sets the “footnote” of the “Page layout composition”.

A.5.22.11 footspace

The **footspace** attribute sets the “footspace” of the “Page layout composition”.

A.5.22.12 headheight

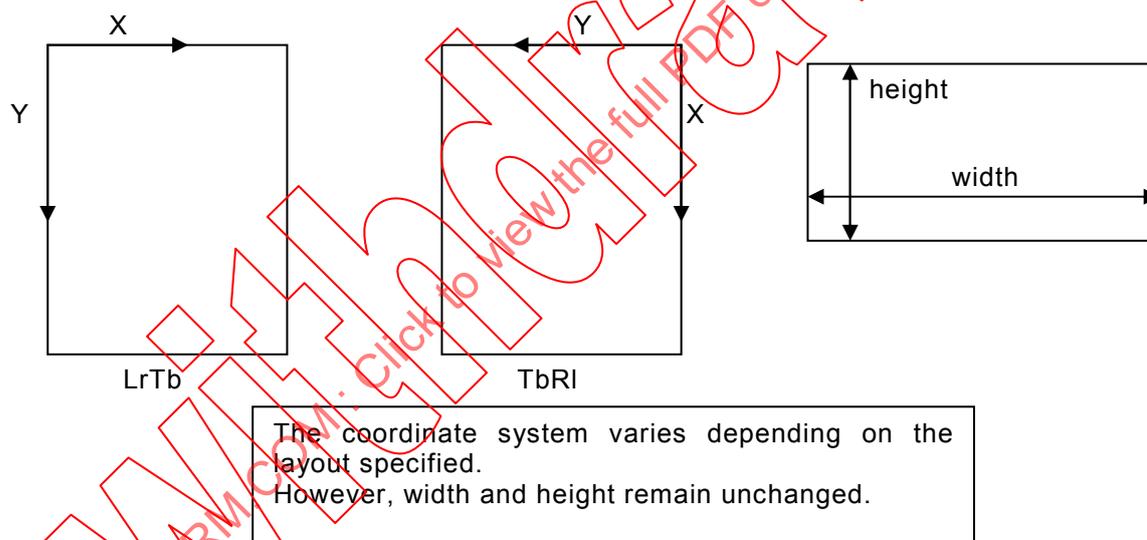
The **headheight** attribute sets the “headheight” of the “Page layout composition”.

A.5.22.13 headsep

The **headsep** attribute sets the “headsep” of the “Page layout composition”.

A.5.22.14 layout

The **layout** attribute specifies the coordinate system of the **Page** element. When “LrTb” is specified, the origin is in the top left, the X-axis is toward the right and the Y-axis is toward the bottom. When “TbRI” is specified, the origin is in the top right, the X-axis is toward the bottom and the Y-axis is toward the left.



IEC 2385/06

Figure A.6 – Layout attribute**A.5.22.15 oddsidemargin**

The **oddsidemargin** attribute sets the left margin of the “Page layout composition” on the 2 screen display: the right page in the horizontal layout, the left page in the vertical layout.

The value of the **oddsidemargin** attribute should be used. On the 1 screen display, the value of “Oddsidemargin” from “Evensidemargin” is used.

A.5.22.16 textheight

The **textheight** attribute sets the “textheight” of the “Page layout composition”.

A.5.22.17 textwidth

The **textwidth** attribute sets the “textwidth” of the “Page layout composition”.

A.5.22.18 topmargin

The **topmargin** attribute sets the “topmargin” of the “Page layout composition”.

A.5.22.19 objid

The **objid** attribute specifies a unique character string in the content for each element.

A.5.23 Wait

The **Wait** element specifies the timer control operation.

Table A.11 – Attribute of Wait

Attribute name	Default value	Data type	Comments
time (required)	(Signed integer)	Decimal string	

The **time** attribute specifies the timer. When the time is “0”, it keeps waiting even for the input operation. When the time is a positive value, nothing is done for the set time × 100 ms). When the time is a negative value, it keeps waiting for the input during the absolute value of the set time (× 100 ms).

A.5.24 TextBlock

The **TextBlock** element specifies the set-up information of the “Block” and “Text” to express the “Block” on the screen. The attribute of the **TextBlock** element is recommended to describe only different information from the **TextStyle** and **BlockStyle** element information specified by “textstyle” and “blockstyle”.

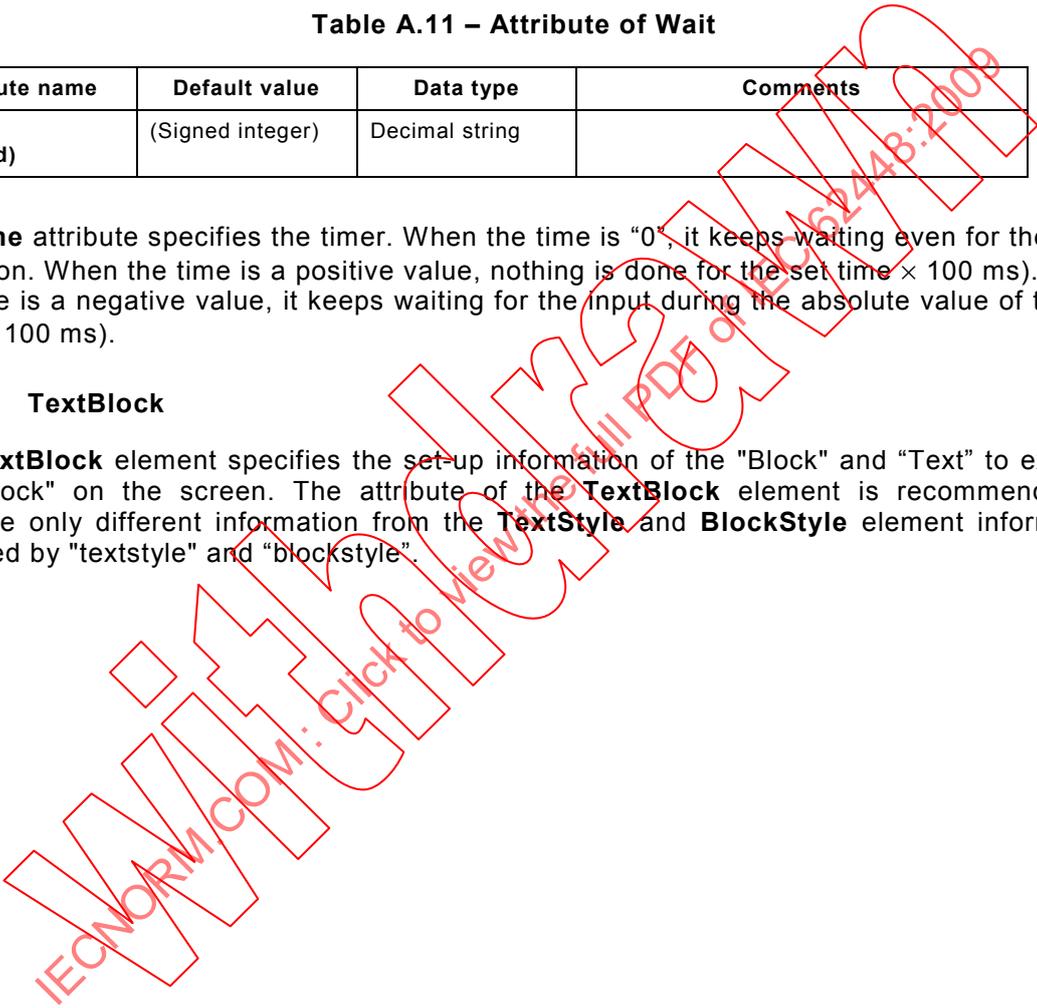


Table A.12 – Attribute of TextBlock

Attribute name	Default value	Data type	Comments
align		String	Specifies "head", "center" or "foot"
baselineskip	(Signed integer)	Decimal string	[pt]*10
bgcolor		COLORREF string	
bgimagemode		String	Specifies "fix", "tile" or "centering". Moreover, it is necessary to specify the refbimage attribute at the same time
blockheight	(Unsigned integer)	Decimal string	[dot]
blockrule		String	Specifies "horz-fixed", "horz-adjustable", "vert-fixed" or "vert-adjustable" as the blockrule attribute of the TextBlock element
blockstyle (Required)		Style string	It is necessary to match the specified style character string to "stylelabel" of the referred BlockStyle element
blockwidth	(Unsigned integer)	Decimal string	[dot]
charspace	(Signed integer)	Decimal string	[pt]*10
column		Decimal string	Specifies an integer between "1" and "9"
columnsep	(Signed integer)	Decimal string	[pt]*10
empdotscode		Hexadecimal string	Specifies the character code used as an emphasis dot. Moreover, it is necessary to specify the empdotsfontname and refempdotsfont attribute at the same time
empdotsposition		String	Specifies "before" or "after"
emplinetype		String	Specifies "none", "solid", "dotted", "double" or "dashed"
emlineposition		String	Specifies "before" or "after"
fontescapement		Decimal string	Specifies "0" or "2 700"
fontfacename		String	Specifies the font name
fontorientation		Decimal string	Specifies "0" or "2 700"
fontsize	(Signed integer)	Decimal string	[pt] * 10
fontweight		Decimal string	Specifies a value between "1" and "1 000"
fontwidth	(Signed integer)	Decimal string	[pt]*10 Specifies "-10" when not changing the font shape
footskip	(Unsigned integer)	Decimal string	[dot]
framecolor		COLORREF string	
framemode		String	Specifies "curve" or "square"
framewidth	(Unsigned integer)	Decimal string	[dot]
layout		String	Specifies "LrTb" or "TbRl"
letterspace	(Signed integer)	Decimal string	[pt]*10
linecolor		COLORREF string	
linespace	(Signed integer)	Decimal string	[pt]*10
textlinewidth	(Unsigned integer)	Decimal string	[pt]*10
objid (Required)		String	Specifies the only character string in the file

Attribute name	Default value	Data type	Comments
parindent	(Signed integer)	Decimal string	[pt]*10
parskip	(Signed integer)	Decimal string	[pt]*10
refbgimage		Object ID string	Specifies the "objid" of the Image element. Moreover, it is necessary to specify the refimagemode attribute at the same time
rubyalign		String	Specifies "start" or "center"
rubyadjust		String	Specifies the "line-edge" or "none" character string
rubyoverhang		String	Specifies the "auto" or "none" character string
sidemargin	(Unsigned integer)	Decimal string	[dot]
textbgcolor		COLORREF string	
textcolor		COLORREF string	
textstyle (Required)		Style string	It is necessary to match the specified style character string to "stylelabel" of the referred TextStyle element
toclabel			Specifies the character string used in the TOC list
topskip	(Unsigned integer)	Decimal string	[dot]
wordspace	(Signed integer)	Decimal string	[pt]*10
empdotsfontname		String	Specifies the font name of the font used for emphasis dots. Moreover, it is necessary to specify the empdotscode and refempdotsfont attribute at the same time
refempdotsfont		Object ID String	Specifies the "objid" of the Font element. Moreover, it is necessary to specify the empdotscode and empdotsname attribute at the same time

A.5.24.1 **fontsize**

The **fontsize** attribute specifies the font size. Units are 10 times pt.

A.5.24.2 **fontwidth**

The **fontwidth** attribute specifies the width of the font. Normally, the optimal width is provided depending on the font size, but this is used when the width is changed intentionally. Units are 10 times pt.

A.5.24.3 **fontescapement**

The **fontescapement** attribute specifies the character feed direction. "0" is specified for horizontal font and "2 700" is specified for vertical font. Other specifications are not accepted.

A.5.24.4 **fontorientation**

The **fontorientation** attribute specifies the direction of character rotation. "0" is specified for horizontal font and "2 700" is specified for vertical font. Other specifications are not accepted.

A.5.24.5 **fontfacename**

The **fontfacename** attribute specifies the name of the font to be used. The standard font is used if nothing is specified.

A.5.24.6 textcolor

The **textcolor** attribute specifies the color of the text to be used. Black (0x00000000) is used if nothing is specified.

A.5.24.7 textbgcolor

The **textbgcolor** attribute specifies the background color of the text. A transparent (0xff000000) background is normally used if nothing is specified.

A.5.24.8 wordspace

The **wordspace** attribute specifies the width of spaces between words of Western languages. Units are 10 times pt.

A.5.24.9 letterspace

The **letterspace** attribute the size of spaces between letters of Western languages. Units are 10 times pt.

A.5.24.10 charspace

The **charspace** attribute specifies the size of spaces between Japanese characters. Units are 10 times pt.

A.5.24.11 baselineskip

The **baselineskip** attribute specifies the space between the lines. Units are 10 times pt.

A.5.24.12 linespace

The **linespace** attribute specifies the minimum guaranteed space for the direction of lines. Units are 10 times pt.

A.5.24.13 parindent

The **parindent** attribute specifies the start position of the first line of the paragraph. A negative value means “overhanging”. The maximum value of “overhanging” is equal to that specified in the **sidemargin** attribute. Units are 10 times pt.

A.5.24.14 parskip

The **parskip** attribute specifies the width of the spaces between paragraphs. Units are 10 times pt.

A.5.24.15 rubyalign

The **rubyalign** attribute specifies the alignment of the “Ruby Text”. The alignment values are specified below.

Table A.13 – rubyalign attribute

Alignment value	
Start aligned	Center aligned
start	center

A.5.24.16 rubyadjust

The **rubyadjust** attribute specifies the operation of the “Ruby Text” at the line edge. The values of the operation are specified below. The adjustment operation is to adjust the head or end position of the “Ruby Text” at the line edge position line break refers to the method used to line up the starting edge or ending edge when the “Ruby Text” is longer than the “Ruby Base”.

Table A.14 – rubyadjust attribute

Line break operation value	
Operation used	Operation not used
line-edge	none

A.5.24.17 rubyoverhang

The **rubyoverhang** attribute specifies whether the overhanging operation for the “Ruby Text” is executed or not, when the “Ruby Text” is longer than the “Ruby Base”. The operation values are specified below.

Table A.15 – rubyoverhang attribute

RubyOverhang	Overhang value
Overhang	auto
No overhang	none

A.5.24.18 empdotsposition

The **empdotsposition** attribute specifies the position of emphasis dots. The values are specified below.

Table A.16 – empdotsposition attribute

EmpDotsPosition	Position value
Before (vertical layout: right, horizontal layout: top)	before
After (vertical layout: left, horizontal layout: under)	after

A.5.24.19 empdotscode

The **empdotscode** attribute specifies the character (or symbol) code used as an emphasis dot.

A.5.24.20 emplineposition

The **emplineposition** attribute specifies the position of the emphasis line. The values are specified below.

Table A.17 – emplineposition attribute

EmpLinePosition	Position value
Before (vertical layout: right, horizontal layout: top)	before
After (vertical layout: left, horizontal layout: under)	after

A.5.24.21 emplinetype

The **emplinetype** attribute specifies the type of emphasis line. The values are specified below.

Table A.18 – emplinetype attribute

Type of line	Line mode value
None	none
Solid line	solid
Dotted line	dotted
Dashed line	dashed
Double line	double

A.5.24.22 column

The **column** attribute specifies the step number of columns.

A.5.24.23 columnsep

The **columnsep** attribute specifies the space between columns. Units are 10 times pt.

A.5.24.24 align

The **align** attribute specifies the alignment direction of the text.

Table A.19 – align attribute

Direction	Value
Head	head
Center	center
Foot	foot

A.5.24.25 textlinewidth

The **textlinewidth** attribute specifies the line width for the **Empline** and **Box** elements. Units are 10 times pt.

A.5.24.26 linecolor

The **linecolor** attribute specifies the line color for the **Empline** and **Box** elements.

A.5.24.27 textstyle

The **textstyle** attribute specifies the “stylelabel” of the **TextStyle** element that specifies the information necessary for rendering the text.

A.5.24.28 fontweight

The **fontweight** attribute specifies the weight of the text. The value “400” is normal weight and “800” is “Bold”.

A.5.24.29 objid

Refer to A.5.22.19.

A.5.24.30 bgcolor

The **bgcolor** attribute specifies the background color of the “Block”.

A.5.24.31 bgimagemode

The **bgimagemode** attribute specifies how the “Image” specified in the **refbgimage** attribute should be positioned.

The value is “fix”: The image position is at the origin of the “Block”.

The value is “tile”: The images fill the “Block” from the origin.

The value is “center”: The image position is in the center of the “Block”.

A.5.24.32 blockheight

The **blockheight** attribute specifies the height of the “Block”.

A.5.24.33 blockrule

The **blockrule** attribute specifies how to expand the “Block” size when the customer is enlarging character size on the viewer devices. The values for the **TextBlock** element are specified “horz-fixed”, “horz-adjustable”, “vert-fixed” and “vert-adjustable”.

Table A.20 – blockrule attribute

Rule value	Rule
horz-fixed	Block width cannot be changed
horz-adjustable	Block width may be changed
vert-fixed	Block height cannot be changed
vert-adjustable	Block height may be changed
block-fixed	Block width and height cannot be changed
block-adjustable	Block width and height may be changed proportionally

A.5.24.34 blockstyle

The **blockstyle** attribute specifies the “stylelabel” of the **BlockStyle** element specifying the information necessary for rendering the text.

A.5.24.35 blockwidth

The **blockwidth** attribute specifies the width of the “Block”.

A.5.24.36 footskip

The **footskip** attribute sets the “footskip” of the “Block layout composition”.

A.5.24.37 framecolor

The **framecolor** attribute specifies the frame color of the “Block”.

A.5.24.38 framemode

The **framemode** attribute specifies the frame type of the “Block”. It is possible to specify two types of frame: “square” means right angles in four corners and “curve” means rounded corners.

A.5.24.39 framewidth

The **framewidth** attribute specifies the frame line width of the “Block”. When this value is “0” or not specified, no frame is rendered.

A.5.24.40 layout

The **layout** attribute specifies the coordinate system of the **TextBlock** element. When “LrTb” is specified, the origin is in the top left, the X-axis is towards the right and the Y-axis is towards the bottom. When “TbRl” is specified, the origin is in the top right, the X-axis is towards the bottom and the Y-axis is towards the left.

A.5.24.41 refbgimage

The **refbgimage** attribute specifies the “objid” of the **Image** element to be shown as the background of the **TextBlock** element. No image is shown if the “objid” is not specified.

A.5.24.42 sidemargin

The **sidemargin** attribute sets the “sidemargin” of the “Block layout composition”.

A.5.24.43 toclabel

The **toclabel** attribute specifies the character string to be shown as the TOC list on the viewer devices. If the “toclabel” is specified, the “Authoring tool” picks up this label’s information and sets this label into the **TOC** element information in auto TOC creation mode.

A.5.24.44 topskip

The **topskip** attribute specifies the “topskip” of the “Block layout composition”.

A.5.24.45 empdotsfontname

The **empdotsfontname** attribute specifies the font name of the character (or symbol) used as emphasis dots.

A.5.24.46 refempdotsfont

The **refempdotsfont** attribute specifies the “objid” of the **Font** element used as emphasis dots. When the value is “0”, the font of the main text is used.

A.5.25 P

The **P** element specifies the minimum unit in the document. In a general novel, this means “Paragraph”. This **P** element is handled as the smallest unit that can be specified as an electronic expression.

Table A.21 – Attribute of P

Attribute name	Default value	Data type	Comments
refesound	0	Decimal string	Specifies the “objid” of the eSound element

The **refesound** attribute specifies the “objid” of the **eSound** element used as the embedded sound playback. If an embedded sound is specified here, the sound is played back at the same time of the **P** element showing.

A.5.26 Plot

The **Plot** element specifies the inline image and button.

Table A.22 – Attribute of Plot

Attribute name	Default value	Data type	Comments
adjustment	"bottom"	String	Specifies "center", "baseline", "top" or "bottom"
refobj (required)		Object ID string	Specifies the "objid" of the Image or Button element
xsize (required)	(Unsigned integer)	Decimal string	[pt]*10
ysize (required)	(Unsigned integer)	Decimal string	[pt]*10

A.5.26.1 xsize

The **xsize** attribute specifies the X direction size of the inline image or inline button. Units are 10 times pt.

A.5.26.2 ysize

The **ysize** attribute specifies the Y direction size of the inline image or inline button. Units are 10 times pt.

A.5.26.3 refobj

The **refobj** attribute specifies the "objid" of the element (the **Image** and **Button** elements) specified as the "inline" target.

A.5.26.4 adjustment

The **adjustment** attribute specifies the position of the inline image or inline button for the text line.

In case of the horizontal layout; The value "top" is to match the top edge of both the inline image and button, and the text line. The value "center" is to match the centerline of both. The value "baseline" means to match the baseline of both. The value "bottom" means to match the bottom edge of both.

In case of the vertical layout; The value "top" is to match the right edge of both the inline image and button, and the text line. The value "center" is to match the centerline of both. The value "baseline" means to match the baseline of both. The value "bottom" means to match the left edge of both.

A.5.27 CR

The **CR** element specifies a compulsory line break. When this is specified immediately after the **P** element, the baseline space is the sum of the "baselineskip" and "parskip".

A.5.28 Fill

The **Fill** element specifies the lead character, which has variable length.

Table A.23 – Attribute of Fill

Attribute name	Default value	Data type	Comments
code (required)		Hexadecimal string	Can be specified between “0x0020” and “0xffff”

The **code** attribute specifies the character code number (UTF-16) that is used in the **Fill** element.

A.5.29 Space

The **Space** element specifies the spacing and kerning between characters.

Table A.24 – Attribute of Space

Attribute name	Default value	Data type	Comments
xsize (required)	(Signed integer)	Decimal string	[dot]

The **xsize** attribute specifies the space in the X-axis direction. The kerning is performed when a negative value is specified.

A.5.30 CharButton

The **CharButton** element specifies the range of the character string to provide the button function.

Table A.25 – Attribute of CharButton

Attribute name	Default value	Data type	Comments
refobj (required)		Object ID string	Specifies the “objid” of the Button element

The **refobj** attribute specifies the “objid” of the element (the **Button** element) specified as the “Button character” target.

A.5.31 Rubi

The **Rubi** element specifies the “Ruby Text” and the “Ruby Base”.

A.5.32 Oyamoji

The **Oyamoji** element specifies the “Ruby Base”.

A.5.33 Gaiji

The **Gaiji** element specifies the character code (UTF-16) of the external character and the alternative character string in case the external character cannot be rendered.

Table A.26 – Attribute of Gaiji

Attribute name	Default value	Data type	Comments
code (required)		Hexadecimal string	Can be specified between “0x0020” and “0xffff”
fontfacename (required)		String	Specifies the font name
refobj (required)		Object ID string	Specifies the “objid” of the Font element

A.5.33.1 refobj

The **refobj** attribute specifies the “objid” of the element (the **Font** element) specified as the “Gaiji character” target.

A.5.33.2 fontfacename

Refer to A.5.24.5.

A.5.33.3 code

The **code** attribute specifies the character code that is used as the external character.

A.5.34 AltString

The **AltString** element specifies the alternative character string. The alternative character string is used when the display device cannot render the original character string.

A.5.35 Org

The **Org** element specifies the original character string that the customers want.

A.5.36 Alt

The **Alt** element specifies the alternative character string instead of the **Org** element.

A.5.37 Rubimoji

The **Rubimoji** element specifies the “Ruby Text” for the “Ruby Base” specified in the **Oyamoji** element.

A.5.38 Box

The **Box** element specifies the character string to be surrounded by a box line.

Table A.27 – Attribute of Box

Attribute name	Default value	Data type	Comments
linetype (required)		String	“solid”, “dotted”, “dashed” or “double” is selected

The **linetype** attribute specifies the line type of the box. The values are specified below.

Table A.28 – linetype attribute

Type of line	Line mode value
Solid line	solid
Dotted line	dotted
Dashed line	dashed
Double line	double

A.5.39 EmpDots

The **EmpDots** element specifies the character string that is added the emphasis dots.

Table A.29 – Attribute of EmpDots

Attribute name	Default value	Data type	Comments
empdotscode		Hexadecimal string	Specifies the character code used as an emphasis dot. Moreover, it is necessary to specify the empdotsfontname and refempdotsfont attribute at the same time
empdotsposition		String	Specifies "before" or "after"
empdotsfontname		String	Specifies the font name used as the emphasis dots. Moreover, it is necessary to specify the empdotscode and refempdotsfont attribute at the same time
refempdotsfont		Object ID string	Specifies the "objid" of the Font element. Moreover, it is necessary to specify the empdotscode and empdotsname attribute at the same time

A.5.39.1 empdotsposition

Refer to A.5.24.18.

A.5.39.2 empdotscode

Refer to A.5.24.19.

A.5.39.3 empdotsfontname

Refer to A.5.24.45.

A.5.39.4 refempdotsfont

Refer to A.5.24.46.

A.5.40 EmpLine

The **EmpLine** element specifies the character string that is added to the emphasis line.

Table A.30 – Attribute of Empline

Attribute name	Default value	Data type	Comments
emplinetype		String	Specifies "none", "solid", "dotted", "dashed" or "double"
emplineposition		String	Specifies "before" or "after"

A.5.40.1 emplineposition

Refer to A.5.24.20.

A.5.40.2 emplinetype

Refer to A.5.24.21.

A.5.41 Sub

The **Sub** element specifies a subscript character string.

A.5.42 Sup

The **Sup** element specifies a superscript character string.

A.5.43 Yoko

The **Yoko** element specifies the character string that has changed the baseline direction to horizontal in the vertical layout.

A.5.44 Tate

The **Tate** element specifies the Western character string that is rotated 90° counter clockwise in the vertical layout.

A.5.45 Nekase

The **Nekase** element specifies the vertical character string that is rotated 90° clockwise in the vertical layout.

A.5.46 NoBR

The **NoBR** element specifies the character string that is not allowed to divide.

A.5.47 DrawChar

The **DrawChar** element specifies the drop cap character.

Table A.31 – Attribute of DrawChar

Attribute name	Default value	Data type	Comments
line (required)	(Unsigned integer)	Decimal string	Specifies the number of lines.

The **line** attribute specifies the number of lines for the characters specified as drop cap.

A.5.48 Italic

The **Italic** element specifies the character string in italics.

A.5.49 Bold

The **Bold** element specifies the character string in bold.

A.5.50 ImageBlock

The **ImageBlock** element specifies the set up information of the "Block" and "Image" to express the "Block" on the screen. The attribute of the **ImageBlock** element is recommended to describe only different information from the **BlockStyle** element information specified by the "blockstyle". The alternative character string is specified in case the image file cannot be opened.

Table A.32 – Attribute of ImageBlock

Attribute name	Default value	Data type	Comments
x0 (required)	(Unsigned integer)	Decimal string	[dot]
y0 (required)	(Unsigned integer)	Decimal string	[dot]
x1 (required)	(Unsigned integer)	Decimal string	[dot]
y1 (required)	(Unsigned integer)	Decimal string	[dot]
xsize (required)	(Unsigned integer)	Decimal string	[dot]
ysize (required)	(Unsigned integer)	Decimal string	[dot]
refstream		Object ID string	Specifies the "objid" of the ImageStream element. Only "refstream" can be selected in the Image element
objid (required)		String	Specifies the only character string in the file
bgcolor		COLORREF string	
bgimagemode		String	Specifies "fix", "tile" or "centering". Moreover, it is necessary to specify the refbimage attribute at the same time
blockheight	(Unsigned integer)	Decimal string	[dot]
blockrule		String	"block-fixed" or "block-adjustable" is specified for the ImageBlock blockrule attribute
blockstyle (required)		Style string	It is necessary to match the specified style character string to "stylelabel" of the referred BlockStyle element
blockwidth	(Unsigned integer)	Decimal string	[dot]
footskip	(Unsigned integer)	Decimal string	[dot]
framecolor		COLORREF string	
framemode		String	Specifies "curve" or "square"
framewidth	(Unsigned integer)	Decimal string	[dot]
layout		String	Specifies "LrTb" or "TbRI"

Attribute name	Default value	Data type	Comments
refbgimage		Object ID string	Specifies the "objid" of the Image element. However, own ID cannot be specified
sidemargin	(Unsigned integer)	Decimal string	[dot]
topskip	(Unsigned integer)	Decimal string	[dot]

A.5.50.1 x0

The **x0** attribute specifies the X-coordinate of the upper left corner required for extracting the portion that needs to be displayed from the source image.

A.5.50.2 y0

The **y0** attribute specifies the Y-coordinate of the upper left corner required for extracting the portion that needs to be displayed from the source image.

A.5.50.3 x1

The **x1** attribute specifies the X-coordinate of the lower right corner required for extracting the portion that needs to be displayed from the source image.

A.5.50.4 y1

The **y1** attribute specifies the Y-coordinate of the lower right corner required for extracting the portion that needs to be displayed from the source image.

A.5.50.5 xsize

The **xsize** attribute specifies the size of the extracted image in the X direction. This refers to expansion along the X-axis if this size is greater than (x1-x0), and contraction along the X-axis if less than (x1-x0).

A.5.50.6 ysize

The **ysize** specifies the size of the extracted image in the Y direction. This refers to expansion along the Y-axis if this size is greater than (y1-y0), and contraction along the Y-axis if less than (y1-y0).

A.5.50.7 refstream

The **refstream** attribute specifies the "objid" of the **ImageStream** element containing the source image data to be displayed.

A.5.50.8 objid

Refer to A.5.22.19.

A.5.50.9 bgcolor

Refer to A.5.24.30.

A.5.50.10 bgimagemode

Refer to A.5.24.31.

A.5.50.11 blockheight

Refer to A.5.24.32.

A.5.50.12 blockrule

The **blockrule** attribute specifies how to expand the “Block” size when the customer is increasing character size on the viewer devices. The values for the **ImageBlock** element are specified as “block-fixed” and “block-adjustable”.

Regarding the table of value, refer to A.5.24.33.

A.5.50.13 blockstyle

Refer to A.5.24.34.

A.5.50.14 blockwidth

Refer to A.5.24.35.

A.5.50.15 footskip

Refer to A.5.24.36.

A.5.50.16 framecolor

Refer to A.5.24.37.

A.5.50.17 framemode

Refer to A.5.24.38.

A.5.50.18 framewidth

Refer to A.5.24.39.

A.5.50.19 layout

The **layout** attribute specifies the coordinate system of the **ImageBlock** element. When “LrTb” is specified, the origin is in the top left, the X-axis is toward the right and the Y-axis is toward the bottom. When “TbRl” is specified, the origin is in the top right, the X-axis is toward the bottom and the Y-axis is toward the left.

A.5.50.20 refbgimage

The **refbgimage** attribute specifies the “objid” of the **Image** element to be shown as the background of the **ImageBlock** element. No image is shown if the “objid” is not specified.

A.5.50.21 sidemargin

Refer to A.5.24.42.

A.5.50.22 topskip

Refer to A.5.24.44.

A.5.51 ButtonBlock

The **ButtonBlock** element specifies the set up information of the "Block" and "Text" to express the "Block" on the screen. The attribute of the **ButtonBlock** element is recommended to describe only different information from the **BlockStyle** element information specified by the "blockstyle". Also the **ButtonBlock** element specifies the action when the button is operated.

There are 4 states in the button operation. Base state(before choice) → Focusin state(choice) → Push state(when push) → Up state(after release). These states are changed in turn by a user operation for the terminal.

Table A.33 – Attribute of ButtonBlock

Attribute name	Default value	Data type	Comments
objid (required)		String	Specifies the only character string in the file
Bgcolor		COLORREF string	
bgimagemode		String	Specifies "fix", "tile" or "centering". Moreover, it is necessary to specify the refbimage attribute at the same time
blockheight	(Unsigned integer)	Decimal string	[dot]
blockrule		String	"block-fixed" or "block-adjustable" is specified for the ButtonBlock blockrule attribute
blockstyle (required)		Style string	It is necessary to match the specified style character string to "stylelabel" of the referred BlockStyle element
blockwidth	(Unsigned integer)	Decimal string	[dot]
footskip	(Unsigned integer)	Decimal string	[dot]
framecolor		COLORREF string	
framemode		String	Specifies "curve" or "square"
framewidth	(Unsigned integer)	Decimal string	[dot]
layout		String	Specifies "LrTb" or "TbRl"
refbimage		Object ID string	Specifies the "objid" of the Image element. Moreover, it is necessary to specify the bgimagemode attribute at the same time
sidemargin	(Unsigned integer)	Decimal string	[dot]
topskip	(Unsigned integer)	Decimal string	[dot]

A.5.51.1 objid

Refer to A.5.22.19.

A.5.51.2 bgcolor

Refer to A.5.24.30.

A.5.51.3 bgimagemode

Refer to A.5.51.3.

A.5.51.4 blockheight

Refer to A.5.24.32.

A.5.51.5 blockrule

The **blockrule** attribute specifies how to expand the “Block” size when the customer is increasing character size on the viewer devices. The values for the **ButtonBlock** element are specified as “block-fixed” and “block-adjustable”.

Regarding the table of the value, refer to A.5.24.33.

A.5.51.6 blockstyle

Refer to A.5.24.34.

A.5.51.7 blockwidth

Refer to A.5.24.35.

A.5.51.8 footskip

Refer to A.5.24.36.

A.5.51.9 framecolor

Refer to A.5.24.37.

A.5.51.10 framemode

Refer to A.5.24.38.

A.5.51.11 framewidth

Refer to A.5.24.39.

A.5.51.12 layout

The **layout** attribute specifies the coordinate system of the **ButtonBlock** element. When “LrTb” is specified, the origin is in the top left, the X-axis is toward the right and the Y-axis is toward the bottom. When “TbRI” is specified, the origin is in the top right, the X-axis is towards the bottom and the Y-axis is towards the left.

A.5.51.13 refbgimage

The **refbgimage** attribute specifies the “objid” of the **Image** element to be shown as the background of the **ButtonBlock** element. No image is shown if the “objid” is not specified.

A.5.51.14 sidemargin

Refer to A.5.24.42.

A.5.51.15 topskip

Refer to A.5.24.44.

A.5.52 BaseButton

The **BaseButton** element specifies the image of the button under the normal state.

Table A.34 – Attribute of BaseButton

Attribute name	Default value	Data type	Comments
refimage		Object ID string	Specifies the "objid" of the Image element

The **refimage** attribute specifies the "objid" of the **Image** element displayed as a button. In case that the button is the CharButton element, an image is not displayed even if specified.

A.5.53 FocusinButton

The **FocusinButton** element specifies the image and the action of the button under the focus-in state. It is possible to specify multiple button operations. These are executed in the sequence specified.

Table A.35 – Attribute of FocusinButton

Attribute name	Default value	Data type	Comments
refimage		Object ID string	Specifies the "objid" of the Image element

Refer to A.5.52.

A.5.54 JumpTo

The **JumpTo** element specifies the movement of the button that jumps to a certain element.

Table A.36 – Attribute of JumpTo

Attribute name	Default value	Data type	Comments
refobj (required)		Object ID string	Specifies the "objid" of TextBlock , ButtonBlock , ImageBlock or Canvas element
refpage (required)		Object ID string	Specifies the "objid" of the Page element

A.5.54.1 refpage

Refer to A.5.20.2.

A.5.54.2 refobj

Refer to A.5.20.1.

A.5.55 Run

The **Run** element specifies the action such as "Window", "Popup window" and "Sound playback".

Table A.37 – Attribute of Run

Attribute name	Default value	Data type	Comments
runoption (required)	"normal"	String	Specifies "normal", "opposite", "center" or "opposite-center"
refobj (required)		Object ID string	Specifies the "objid" of the PopUpWin , Window or Sound element

A.5.55.1 runoption

The **runoption** attribute specifies the display position of Window and Pop Up Window in the 2 screen display. The values are specified below.

Table A.38 – runoption attribute

Position	Position option
Button side screen	normal
Opposite button side screen	opposite
Center of button side screen	center
Center of opposite button side screen	opposite-center

A.5.55.2 refobj

The **refobj** attribute specifies the “objid” of the element (the **Window**, **PopUpWin** or **Sound** element) specified as the “Run action” target.

A.5.56 SoundStop

The **SoundStop** element forcibly stops the asynchronous sound playback.

A.5.57 CloseWindow

The **CloseWindow** element forcibly closes the window being displayed.

A.5.58 PushButton

The **PushButton** element specifies the image and the action of the button under the button-push state. It is possible to specify multiple button operations. These are executed in the sequence specified.

Table A.39 – Attribute of PushButton

Attribute name	Default value	Data type	Comments
refimage		Object ID string	Specifies the “objid” of the Image element

Refer to A.5.52.

A.5.59 UpButton

The **UpButton** element specifies the image and the action of the button under the button-up state. It is possible to specify multiple button operations. These are executed in the sequence specified. Regarding the **PopUpWin** element, it cannot be invoked by the **Run** element.

Table A.40 – Attribute of UpButton

Attribute name	Default value	Data type	Comments
refimage		Object ID string	Specifies the “objid” of the Image element

Refer to A.5.52.

A.5.60 Canvas

The **Canvas** element specifies the absolute coordinates area to create the complex layout.

Table A.41 – Attribute of Canvas

Attribute name	Default value	Data type	Comments
bgcolor	"0xff000000"	COLORREF string	
blockrule	"block-fixed"	String	Specifies "block-fixed" or "block-adjustable" as the blockrule attribute value of the Canvas element
canvasheight	"0" (Unsigned integer)	Decimal string	[dot]
canvaswidth	"0" (Unsigned integer)	Decimal string	[dot]
framecolor	"0x00000000"	COLORREF string	
framewidth	"0" (Unsigned integer)	Decimal string	[dot]
layout	"LrTb"	String	Specifies "LrTb" or "TbRl"
objid (Required)		String	Specifies the only character string in the file
toclabel		String	Specifies the character string used in the TOC list
framemode	"square"	String	Specifies "curve" or "square"

A.5.60.1 canvaswidth

The **canvaswidth** attribute specifies the width of the **Canvas** element area.

A.5.60.2 canvasheight

The **canvasheight** attribute specifies the height of the **Canvas** element area.

A.5.60.3 bgcolor

The **bgcolor** attribute specifies the background color of the "the Canvas element area".

A.5.60.4 layout

The **layout** attribute specifies the coordinate system of the **Canvas** element. When "LrTb" is specified, the origin is in the top left, the X-axis is towards the right and the Y-axis is towards the bottom. When "TbRl" is specified, the origin is in the top right, the X-axis is towards the bottom and the Y-axis is towards the left.

A.5.60.5 framewidth

The **framewidth** attribute specifies the frame line width of the "Canvas". When this value is "0" or not specified, no frame is rendered.

A.5.60.6 framecolor

The **framecolor** attribute specifies the frame color of the "Canvas".

A.5.60.7 blockrule

The **blockrule** attribute specifies how to expand the "Block" size when the customer is increasing character size on the viewer devices. The values for the **Canvas** element are specified as "block-fixed" and "block-adjustable".

Regarding the table of the value, refer to A.5.24.33.

A.5.60.8 objid

Refer to A.5.22.19.

A.5.60.9 toclabel

Toclabel assigns an item name for the "Canvas" for use in the table of contents list. The **toclabel** attribute specifies the character string to be shown as the TOC list on the viewer devices. If the "toclabel" is specified, the "Authoring tool" picks up this label's information and sets this label into the **TOC** element information in auto TOC creation mode.

A.5.60.10 framemode

The **framemode** attribute specifies the frame type of the "Canvas". It is possible to specify two types of frame: "square" means right angles in four corners and "curve" means rounded corners.

A.5.61 PutObj

The **PutObj** element specifies the position of elements on the **Canvas** element.

Table A.42 – Attribute of PutObj

Attribute name	Default value	Data type	Comments
refobj (required)		Object ID string	Specifies the "objid" of the PopUpWin , TextBlock , ImageBlock , ButtonBlock or Sound element. Further, when used in the Header or Footer element, TextBlock and ImageBlock can be specified.
x1 (required)	(Unsigned integer)	Decimal string	[dot]
y1 (required)	(Unsigned integer)	Decimal string	[dot]

A.5.61.1 x1

The **x1** attribute specifies the X-axis coordinates for locating the origin point of the "objid" element specified with refobj.

A.5.61.2 y1

The **y1** attribute specifies the Y-axis coordinates for locating the origin point of the "objid" element specified with refobj.

A.5.61.3 refobj

The **refobj** attribute specifies the "objid" of the element specified as the shown element.

A.5.62 Moveto

The **Moveto** element specifies the next start position of the **Lineto**, **DrawBox** and **DrawEllipse** in the **Canvas** element.

Table A.43 – Attribute of Moveto

Attribute name	Default value	Data type	Comments
x1 (required)	(Signed integer)	Decimal string	[dot]
y1 (required)	(Signed integer)	Decimal string	[dot]

A.5.62.1 x1

The **x1** attribute specifies the X-coordinate of the drawing start point.

A.5.62.2 y1

The **y1** attribute specifies the Y-coordinate of the drawing start point.

A.5.63 Lineto

The **Lineto** element specifies a straight line drawing in the **Canvas** element. The straight line is drawn from the start point to the position (end point) specified by the **Lineto** element. After the line has been drawn, the specified drawing end point becomes the next drawing start point.

Table A.44 – Attribute of Lineto

Attribute name	Default value	Data type	Comments
linecolor		COLORREF string	
linetype		String	Specifies "solid", "dotted", "dashed", or "double"
arrowtype		String	Specifies "none", "begin", "end", or "both"
linewidth	(Unsigned integer)	Decimal string	[dot]
x1 (required)	(Signed integer)	Decimal string	[dot]
y1 (required)	(Signed integer)	Decimal string	[dot]

A.5.63.1 x1

The **x1** attribute specifies the X-coordinate of the drawing end point.

A.5.63.2 y1

The **y1** attribute specifies the Y-coordinate of the drawing end point.

A.5.63.3 linewidth

The **linewidth** attribute specifies the width of the line to be drawn.

A.5.63.4 linecolor

The **linecolor** attribute specifies the line color for the **Lineto** element.

A.5.63.5 linetype

The **linetype** attribute specifies the line type to be drawn.

Regarding the table of the value, refer to A.5.38.

A.5.63.6 arrowtype

The **arrowtype** attribute specifies whether the line drawn in the **Canvas** element is a straight line or an arrow.

Table A.45 – arrowtype attribute

Arrow type	Value
Straight line	none
Start point arrow	begin
End point arrow	end
Both end arrows	both

A.5.64 DrawBox

The **DrawBox** element specifies a rectangular drawing in the **Canvas** element. The quadrangle is drawn as the diagonal from the start point to the end point specified by the **DrawBox** element.

Table A.46 – Attribute of DrawBox

Attribute name	Default value	Data type	Comments
fillcolor		COLORREF string	
linecolor		COLORREF string	
linetype		String	Specifies "solid", "dotted", "dashed", or "double"
linewidth	(Unsigned integer)	Decimal string	[dot]
x1 (required)	(Signed integer)	Decimal string	[dot]
y1 (required)	(Signed integer)	Decimal string	[dot]

A.5.64.1 x1

Refer to A.5.63.1.

A.5.64.2 y1

Refer to A.5.63.2.

A.5.64.3 linewidth

The **linewidth** attribute specifies the line width of the rectangle to be drawn.

A.5.64.4 linecolor

The **linecolor** attribute specifies the line color for the **DrawBox** element.

A.5.64.5 linetype

The **linetype** attribute specifies the line type of the rectangle to be drawn. The values are specified below.

Regarding the table of the value, refer to A.5.38.

A.5.64.6 fillcolor

The **fillcolor** attribute fills the body color of the rectangle to be drawn.

A.5.65 DrawEllipse

The **DrawEllipse** element specifies an ellipse drawing in the **Canvas** element. The quadrangle is drawn as the diagonal from the start point to the end point specified by the **DrawEllipse** element.

Table A.47 – Attribute of DrawEllipse

Attribute name	Default value	Data type	Comments
fillcolor		COLORREF string	
linecolor		COLORREF string	
linetype		string	Specifies "solid", "dotted", "dashed", or "double"
linewidth	(Unsigned integer)	Decimal string	[dot]
x1 (required)	(Signed integer)	Decimal string	[dot]
y1 (required)	(Signed integer)	Decimal string	[dot]

A.5.65.1 x1

Refer to A.5.63.1.

A.5.65.2 y1

Refer to A.5.63.2.

A.5.65.3 linewidth

The **linewidth** attribute specifies the line width of the ellipse to be drawn.

A.5.65.4 linecolor

The **linecolor** attribute specifies the line color for the **DrawEllipse** element.

A.5.65.5 linetype

The **linetype** attribute specifies the line type of the ellipse to be drawn. The values are specified below.

Regarding the table of the value, refer to A.5.38.

A.5.65.6 fillcolor

The **fillcolor** attribute fills the body color of the ellipse to be drawn.

A.5.66 RuledLine

The **RuledLine** element specifies the ruled lines.

Table A.48 – Attribute of RuledLine

Attribute name	Default value	Data type	Comments
linelength (required)	(Unsigned integer)	Decimal string	[dot]
linetype (required)		string	Specifies “solid”, “dotted”, “dashed”, or “double”
linewidth (required)	(Unsigned integer)	Decimal string	[dot]
linecolor	0x00000000	COLORREF string	

A.5.66.1 linewidth

The **linewidth** attribute specifies the width of the ruled line.

A.5.66.2 linelength

The **linelength** attribute specifies the length of the ruled line. If the length is not specified, the ruled line will be drawn with its length extending through the entire available drawing area.

A.5.66.3 linetype

The **linetype** attribute specifies the type of the rule line. The values are specified below.

Regarding the table of the value, refer to A.5.38.

A.5.66.4 linecolor

The **linecolor** attribute specifies the line color for the **RuledLine** element.

A.5.67 Style

The **Style** element specifies each type of Style information.

A.5.68 BookStyle

The **BookStyle** element specifies the content information.

Table A.49 – Attribute of BookStyle

Attribute name	Default value	Data type	Comments
stylelabel (required)		Style string	Specifies the style character string
objid (required)		String	Specifies the only character string in the file

A.5.68.1 stylelabel

The **stylelabel** attribute specifies an identifiable name of the **Style** element.

A.5.68.2 objid

Refer to A.5.22.19.

A.5.69 SetDefault

The **SetDefault** element specifies the default value of the available layout information in the content. The value set here will be the default value for using each type of object.

Table A.50 – Attribute of SetDefault

Attribute name	Default value	Data type	Comments
rubyalign (required)	"start"	String	Specifies the "start" or "center" character string
rubyadjust (required)	"none"	String	Specifies the "line-edge" or "none" character string
rubyoverhang (required)	"none"	String	Specifies the "auto" or "none" character string
empdotsposition (required)	"before"	String	Specifies "before" or "after"
emplineposition (required)	"before"	String	Specifies "before" or "after"
emplinetype (required)	"none"	Hexadecimal string	Specifies the line mode value as "none", "solid", "dotted", "dashed", or "double"
empdotsfontname (required)		String	Specifies the font name of the font used for emphasis dots. Moreover, it is necessary to specify the empdotscode and refempdotsfont attribute at the same time
refempdotsfont (required)		Object ID string	Specifies the "objid" of the Font element. Moreover, it is necessary to specify the empdotscode and empdotsfontname attribute at the same time
empdotscode (required)	0x3001	Hexadecimal string	Specifies the character code used as an emphasis dot. Moreover, it is necessary to specify the empdotsfontname and refempdotsfont attribute at the same time
setwaitpropt (required)	"noreplay"	String	Specifies "replay" or "noreplay"

A.5.69.1 rubyalign

Refer to A.5.24.15.

A.5.69.2 rubyadjust

Refer to A.5.24.16.

A.5.69.3 rubyoverhang

Refer to A.5.24.17.

A.5.69.4 empdotsposition

Refer to A.5.24.18.

A.5.69.5 empdotscode

Refer to A.5.24.19.

A.5.69.6 emplineposition

Refer to A.5.24.20.

A.5.69.7 emplinetype

Refer to A.5.24.21.

A.5.69.8 empdotsfontname

Refer to A.5.24.45.

A.5.69.9 refempdotsfont

Refer to A.5.24.46.

A.5.69.10 setwaitprop

The **setwaitprop** attribute specifies the “Wait” process within the **Page** element. The following table is used to specify whether the process is replayed.

Table A.51 – setwaitprop attribute

Wait flag value	Operation specification
replay	Replay
noreplay	No replay

A.5.70 RegistFont

The **RegistFont** element specifies the font to be opened in advance.

Table A.52 – Attribute of RegistFont

Attribute name	Default value	Data type	Comments
fontfilename (required)		String	Specifies the after-open file name
file (required)		File path string	Describes with the file's absolute path or accessible relative path
encoding (required)		String	Specifies “TTF”, “OTF”, or “BF”
fontname (required)		String	Specifies the font name

A.5.70.1 fontfilename

The **fontfilename** attribute specifies the font file name to be used by the display software when the font is opened.

A.5.70.2 file

The **file** attribute specifies the path for the font file to be embedded, and its file name.

A.5.70.3 fontname

The **fontname** attribute specifies the font name of the font to be embedded.

A.5.70.4 encoding

The **encoding** attribute specifies the data format of the font file. Character strings that can be described are “TTF”, “OTF”, and “BF”. However, in this version, “OTF” and “BF” are reserved words, and cannot actually be set.

A.5.71 BookSetting

The **BookSetting** element specifies the assumed environment when the content was created.

Table A.53 – Attribute of BookSetting

Attribute name	Default value	Data type	Comments
bindingdirection (required)		String	Specifies "Lr" or "Rl"
dpi (required)	(Unsigned integer)	Decimal string	[dpi]*10
screenheight (required)	(Unsigned integer)	Decimal string	[dot]
screenwidth (required)	(Unsigned integer)	Decimal string	[dot]
Colordepth	"24" (Unsigned integer)	Decimal string	Specifies the color depth in bits

A.5.71.1 bindingdirection

The **bindingdirection** attribute specifies the page flow direction of the content. If "Lr" is specified, the page advances from left to right, and if "Rl" is specified, the page advances from right to left.

A.5.71.2 dpi

The **dpi** attribute specifies the assumed dpi value when the layout of the content was determined.

A.5.71.3 screenwidth

The **screenwidth** attribute specifies the assumed width of the display area when the layout of the content was determined.

A.5.71.4 screenheight

The **screenheight** attribute specifies the assumed height of the display area when the layout of the content was determined.

A.5.71.5 colordepth

The **colordepth** attribute specifies the bits the color depth required to display the content as it is.

A.5.72 TextStyle

The **TextStyle** element specifies the available "Text" information in the content.

Table A.54 – Attribute of TextStyle

Attribute name	Default value	Data type	Comments
objid (required)		String	Specifies the only character string in the file
stylelabel (required)		Style string	Specifies the style character string
Fontsize	"0" (Signed integer)	Decimal string	[pt]*10
Fontwidth	"-10" (Signed integer)	Decimal string	[pt]*10 Specifies "-10" when not changing the font shape
Fontescapement	"0"	Decimal string	Specifies "0" or "2700"
Fontorientation	"0"	Decimal string	Specifies "0" or "2700"
fontfacename (required)		String	Specifies the font name
Textcolor	"0x00000000"	COLORREF string	
Textbgcolor	"0xff000000"	COLORREF string	
Wordspace	"0" (Signed integer)	Decimal string	[pt]*10
Letterspace	"0" (Signed integer)	Decimal string	[pt]*10
Baselineskip	"0" (Signed integer)	Decimal string	[pt]*10
Linespace	"0" (Signed integer)	Decimal string	[pt]*10
Parindent	"0" (Signed integer)	Decimal string	[pt]*10
Parskip	"0" (Signed integer)	Decimal string	[pt]*10
Rubyalign		String	Specifies the "start" or "center" character string
Rubyadjust		String	Specifies the "line-edge" or "none" character string
Rubyoverhang		String	Specifies the "auto" or "none" character string
Empdotsposition		String	Specifies "before" or "after"
Empdotscode		String	Specifies the character code used as an emphasis dot. Moreover, it is necessary to specify the empdotsfontname and refempdotsfont attribute at the same time
Emlineposition		String	Specifies "before" or "after"
Emlinetype		String	Specifies the line mode value as "none", "solid", "dotted", "dashed", or "double"
Column	"1"	Decimal string	Specifies an integer value from "1" to "9"
Columnsep	"0"	Decimal string	[pt]*10
Align	"head"	String	Specifies "head", "center", or "foot"
Textlinewidth	"0" (Unsigned integer)	Decimal string	[pt]*10
Linecolor	"0x00000000"	COLORREF string	

Attribute name	Default value	Data type	Comments
Charspace	"0" (Signed integer)	Decimal string	[pt]*10
Fontweight	"400"	Decimal string	Specifies a value from "1" to "1,000"
empdotsfontname		String	Specifies the font name of the font used for emphasis dots. Moreover, it is necessary to specify the empdotscode and refempdotsfont attribute at the same time
Refempdotsfont		Object ID string	Specifies the "objid" of the Font element. Moreover, it is necessary to specify the empdotscode and empdotsfontname attribute at the same time

A.5.72.1 objid

Refer to A.5.22.19.

A.5.72.2 stylelabel

The **stylelabel** attribute specifies an identifiable name of the **Style** element. It is necessary to match it to "textstyle" attribute name of the **TextBlock** element so that it is referred.

A.5.72.3 fontsize

Refer to A.5.24.1.

A.5.72.4 fontwidth

Refer to A.5.24.2.

A.5.72.5 fontescapement

Refer to A.5.24.3.

A.5.72.6 fontorientation

Refer to A.5.24.4.

A.5.72.7 fontfacename

Refer to A.5.24.5.

A.5.72.8 textcolor

Refer to A.5.24.6.

A.5.72.9 textbgcolor

Refer to A.5.24.7.

A.5.72.10 wordspace

Refer to A.5.24.8.

A.5.72.11 letterspace

Refer to A.5.24.9.

A.5.72.12 baselineskip

Refer to A.5.24.11.

A.5.72.13 linespace

Refer to A.5.24.12.

A.5.72.14 parindent

Refer to A.5.24.13.

A.5.72.15 parskip

Refer to A.5.24.14.

A.5.72.16 rubyalign

Refer to A.5.24.15.

A.5.72.17 rubyadjust

Refer to A.5.24.16.

A.5.72.18 rubyoverhang

Refer to A.5.24.17.

A.5.72.19 empdotsposition

Refer to A.5.24.18.

A.5.72.20 empdotscode

Refer to A.5.24.19.

A.5.72.21 emplineposition

Refer to A.5.24.20.

A.5.72.22 emplinetype

Refer to A.5.24.21.

A.5.72.23 column

Refer to A.5.24.22.

A.5.72.24 columnsep

Refer to A.5.24.23.

A.5.72.25 align

Refer to A.5.24.24.

A.5.72.26 textlinewidth

Refer to A.5.24.25.

A.5.72.27 linecolor

Refer to A.5.24.26.

A.5.72.28 charspace

Refer to A.5.24.10.

A.5.72.29 fontweight

Refer to A.5.24.28.

A.5.72.30 empdotsfontname

Refer to A.5.24.45.

A.5.72.31 refempdotsfont

Refer to A.5.24.46.

A.5.73 BlockStyle

The **BlockStyle** element specifies the available “Block” information in the content.

Table A.55 – Attribute of BlockStyle

Attribute name	Default value	Data type	Comments
stylelabel (required)		Style string	Specifies a style character string
objid (required)		String	Specifies the only character string in the file
blockwidth	“0” (Unsigned integer)	Decimal string	[dot]
blockheight	“0” (Unsigned integer)	Decimal string	[dot]
blockrule	“horz-adjustable”	String	Specifies “horz-fixed”, “horz-adjustable”, “vert-fixed”, “vert-adjustable”, “block-fixed”, or “block-adjustable”
bgcolor	“0xff000000”	COLORREF string	
layout	“LrTb”	String	Selects “LrTb” or “TbRl”
framewidth	“0” (Unsigned integer)	Decimal string	[dot]
framecolor	“0x00000000”	COLORREF string	
framemode	“square”	String	Specifies “curve” or “square”
topskip	“0” (Unsigned integer)	Decimal string	[dot]
sidemargin	“0” (Unsigned integer)	Decimal string	[dot]
footskip	“0” (Unsigned integer)	Decimal string	[dot]
refbgimage	“0”	Object ID string	Specifies the “objid” of the Image element. Moreover, it is necessary to specify the bgimagemode attribute at the same time
bgimagemode	“fix”	String	Specifies “fix”, “tile” or “centering”. Moreover, it is necessary to specify the refbgimage attribute at the same time

A.5.73.1 stylelabel

The **stylelabel** attribute specifies an identifiable name of the **Style** element. It is necessary to match it to "blockstyle" attribute name of the **TextBlock**, **ImageBlock** and **ButtonBlock** element so that it is referred.

A.5.73.2 objid

Refer to A.5.22.19.

A.5.73.3 blockwidth

Refer to A.5.24.35.

A.5.73.4 blockheight

Refer to A.5.24.32.

A.5.73.5 blockrule

The **blockrule** attribute specifies how to expand the "Block" size when the customer is increasing character size on the viewer devices. The values for the **TextBlock**, **ImageBlock** and **ButtonBlock** elements are specified as "block-fixed" and "block-adjustable".

Regarding the table of the value, refer to A.5.24.33.

A.5.73.6 bgcolor

Refer to A.5.24.30.

A.5.73.7 layout

The **layout** attribute specifies the coordinate system of the **TextBlock**, **ImageBlock** and **ButtonBlock** elements. When "LrTb" is specified, the origin is in the top left, the X-axis is towards the right and the Y-axis is towards the bottom. When "TbRI" is specified, the origin is in the top right, the X-axis is towards the bottom and the Y-axis is towards the left.

A.5.73.8 framewidth

Refer to A.5.24.39.

A.5.73.9 framecolor

Refer to A.5.24.37.

A.5.73.10 framemode

Refer to A.5.24.38.

A.5.73.11 topskip

Refer to A.5.24.44.

A.5.73.12 sidemargin

Refer to A.5.24.42.

A.5.73.13 footskip

Refer to A.5.24.36.

A.5.73.14 refbgimage

The **refbgimage** attribute specifies the “objid” of the **Image** element to be shown as the background of the **BlockStyle** element. No image is shown if the “objid” is not specified.

A.5.73.15 bgimagemode

Refer to A.5.73.15.

A.5.74 PageStyle

The **PageStyle** element specifies the available “Page” information in the content.

Table A.56 – Attribute of PageStyle

Attribute name	Default value	Data type	Comments
stylelabel (required)		Style string	Specifies a style character string
objid (required)		String	Specifies the only character string in the file
Refbgimage	“0”	Object ID string	Specifies the “objid” of the Image element. Moreover, it is necessary to specify the bgimagemode attribute at the same time
Bgimagemode	“fix”	String	Specifies “fix”, “tile” or “centering”. Moreover, it is necessary to specify the refbgimage attribute at the same time
Evenfooterid	“0”	Object ID string	Specifies the “objid” of the Footer element
Evenheaderid	“0”	Object ID string	Specifies the “objid” of the Header element
Oddfooterid	“0”	Object ID string	Specifies the “objid” of the Footer element
Oddheaderid	“0”	Object ID string	Specifies the “objid” of the Header element
Pageposition	“any”	String	Specifies “upper”, “lower”, or “any”
Setemptyview	“show”	String	Specifies “show” or “empty”
Setwaitprop		String	Specifies “replay” or “noreplay”
Topmargin	“0” (Signed integer)	Decimal string	[dot]
Headheight	“0” (Signed integer)	Decimal string	[dot]
Headsep	“0” (Signed integer)	Decimal string	[dot]
oddsidemargin	“0” (Unsigned integer)	Decimal string	[dot]
evensidemargin	“0” (Unsigned integer)	Decimal string	[dot]
Textheight	“0” (Unsigned integer)	Decimal string	[dot]
Textwidth	“0” (Unsigned integer)	Decimal string	[dot]

Attribute name	Default value	Data type	Comments
footheight	"0" (Signed integer)	Decimal string	[dot]
Layout	"LrTb"	String	Specifies "LrTb" or "TbRl"
Footspace	"0" (Signed integer)	Decimal string	[dot]

A.5.74.1 stylelabel

The **stylelabel** attribute specifies an identifiable name of the **Style** element. It is necessary to match it to "pagestyle" attribute name of the **Page** element so that it is referred.

A.5.74.2 objid

Refer to A.5.22.19.

A.5.74.3 refbgimage

The **refbgimage** attribute specifies the "objid" of the **Image** element to be shown as the background of the **PageStyle** element. No image is shown if the "objid" is not specified.

A.5.74.4 bgimagemode

Refer to A.5.22.8.

A.5.74.5 evenfooterid

Refer to A.5.22.7.

A.5.74.6 evenheaderid

Refer to A.5.22.5.

A.5.74.7 oddfooterid

Refer to A.5.22.6.

A.5.74.8 oddheaderid

Refer to A.5.22.4.

A.5.74.9 pageposition

Refer to A.5.22.3.

A.5.74.10 setemptyview

The **setemptyview** attribute specifies whether to display the "View" with the "Header" and "Footer" or a completely empty "View" for a blank page on the screen.

The "Header" and "Footer" are shown when "show" is specified, and a completely empty "View" is shown when "empty" is specified.

A.5.74.11 setwaitprop

Refer to A.5.69.10.

A.5.74.12 topmargin

Refer to A.5.22.18.

A.5.74.13 headheight

Refer to A.5.22.12.

A.5.74.14 headsep

Refer to A.5.22.13.

A.5.74.15 oddsidemargin

Refer to A.5.22.15.

A.5.74.16 evensidemargin

Refer to A.5.22.9.

The value of the **evensidemargin** attribute should be used.

A.5.74.17 textheight

Refer to A.5.22.16.

A.5.74.18 textwidth

Refer to A.5.22.17.

A.5.74.19 footspace

Refer to A.5.22.11.

A.5.74.20 footheight

Refer to A.5.22.10.

A.5.74.21 layout

Refer to A.22.14.

A.5.75 Objects

The **Objects** element specifies the elements that are used repeatedly and the independent elements of the flow of the **Main** element, such as windows, sounds, etc. The order of description of each element is free in the “Objects”.

A.5.76 Window

The **Window** element specifies the area (“Window”) of one high hierarchy from the screen on the main text display. It is possible to compose the layout of the character and the image in the **Window** element.

The “Window” can be eliminated by performing the “close” operation, this is the different point from “PopUpWindow” described later.

Table A.57 – Attribute of Window

Attribute name	Default value	Data type	Comments
windowwidth	"0" (Unsigned integer)	Decimal string	[dot]
windowheight	"0" (Unsigned integer)	Decimal string	[dot]
Layout	"LrTb"	String	Specifies the "LrTb" or "TbRI" character string
objid (required)		String	Specifies the only character string in the file

A.5.76.1 windowwidth

The **windowwidth** attribute specifies the width of the the **Window** element.

A.5.76.2 windowheight

The **windowheight** attribute specifies the height of the **Window** element.

A.5.76.3 layout

The **layout** attribute specifies the coordinate system of the **Window** element. When "LrTb" is specified, the origin is in the top left, the X-axis is towards the right and the Y-axis is towards the bottom. When "TbRI" is specified, the origin is in the top right, the X-axis is towards the bottom and the Y-axis is towards the left.

A.5.76.4 objid

Refer to A.5.22.19.

A.5.77 PopUpWin

The **PopUpWin** element specifies the area ("PopUpWindow") of one high hierarchy from the screen on the main text display. It is possible to describe the character and the image in the **PopUpWin** element.

Table A.58 – Attribute of PopUpWin

Attribute name	Default value	Data type	Comments
objid (required)		String	Specifies the only character string in the file
popupwinlabel (required)		String	

A.5.77.1 objid

Refer to A.5.22.19.

A.5.78 Sound

The **Sound** element specifies the information of sounds to be played back.

Table A.59 – Attribute of Sound

Attribute name	Default value	Data type	Comments
times	"1"	Decimal string	Specifies the number of times sounds will be played. "0" specifies an unlimited number of times
playmode	"sync"	String	Specifies "sync" or "async"
refstream		Object ID string	Specifies the "objid" of the SoundStream element. Moreover, it is necessary to specify the refstream attribute at the same time
objid (required)		String	Specifies the only character string in the file

A.5.78.1 times

The **times** attribute specifies the number of times sounds will be played.

A.5.78.2 playmode

The **playmode** attribute specifies the method of playing sounds. It specifies "sync" for synchronous play, or "async" for asynchronous play.

A.5.78.3 refstream

The **refstream** attribute specifies the "objid" of the **SoundStream** element that specifies the sound source data to be played.

A.5.78.4 objid

Refer to A.5.22.19.

A.5.79 SoundStream

The **SoundStream** element specifies the source data of the sound.

Table A.60 – Attribute of SoundStream

Attribute name	Default value	Data type	Comments
encoding (required)		String	Specifies "PCM" or "MP3"
file (required)		File path	Specifies the file path and file name that contain the data
objid (required)		String	Specifies the only character string in the file

A.5.79.1 encoding

The **encoding** attribute specifies the data format of the sound file. The character strings that can be described are "PCM" and "MP3".

A.5.79.2 file

The **file** attribute specifies the path and file name of the sound data.

A.5.79.3 objid

Refer to A.5.22.19.

A.5.80 ImageStream

The **ImageStream** element specifies the source data of the image.

Table A.61 – Attribute of ImageStream

Attribute name	Default value	Data type	Comments
encoding (required)		String	Specifies "JPEG", "GIF", "BMP", or "PNG"
file (required)		File path	Specifies the file path and file name that contain the data
objid (required)		String	Specifies the only character string in the file

A.5.80.1 encoding

The **encoding** attribute specifies the data format of the image file. The character strings that can be described are "JPEG", "GIF", "BMP", and "PNG".

A.5.80.2 file

The **file** attribute specifies the path and file name of the image data.

A.5.80.3 objid

Refer to A.5.22.19.

A.5.81 Header

The **Header** element specifies the information to be displayed in the header. The header area is determined by the "textwidth" and "headheight" of the **Page** element.

Table A.62 – Attribute of Header

Attribute name	Default value	Data type	Comments
Layout	"LrTb"	String	Specifies "LrTb" or "TbRl"
Bgcolor	"0xff00000"	COLORREF string	
Framewidth	"0" (Signed integer)	Decimal String	[dot]
Framecolor	"0x00000000"	COLORREF string	
Framemode	"square"		Specifies "curve" or "square"
objid (Required)		String	Specifies the only character string in the file

A.5.81.1 layout

The **layout** attribute specifies the coordinate system of the **Header** element. When "LrTb" is specified, the origin is in the top left, the X-axis is towards the right and the Y-axis is towards the bottom. When "TbRl" is specified, the origin is in the top right, the X-axis is towards the bottom and the Y-axis is towards the left.

A.5.81.2 bgcolor

The **bgcolor** attribute specifies the background color of the "Header".

A.5.81.3 framewidth

The **framewidth** attribute specifies the frame line width of the “Header”. When this value is “0” or not specified, no frame is rendered.

A.5.81.4 framecolor

The **framecolor** attribute specifies the frame color of the “Header”.

A.5.81.5 framemode

The **framemode** attribute specifies the type of frame surrounding the header area. It is possible to specify two types of frame: “square” means right angles in four corners and “curve” means rounded corners.

A.5.81.6 objid

Refer to A.5.22.19.

A.5.82 Footer

The **Footer** element specifies the information to be displayed in the footer. The footer area is determined by the “textwidth” and “footheight” of the **Page** element.

Table A.63 – Attribute of Footer

Attribute name	Default value	Data type	Comments
Layout	“LrTb”	String	Specifies “LrTb” or “TbRl”
Bgcolor	“0xff000000”	COLORREF string	
Framewidth	“0” (Signed integer)	Decimal String	[dot]
Framecolor	“0x00000000”	COLORREF string	
Framemode	“square”	String	Specifies “curve” or “square”
objid (required)		String	Specifies the only character string in the file

A.5.82.1 layout

The **layout** attribute specifies the coordinate system of the **Footer** element. When “LrTb” is specified, the origin is in the top left, the X-axis is towards the right and the Y-axis is towards the bottom. When “TbRl” is specified, the origin is in the top right, the X-axis is towards the bottom and the Y-axis is towards the left.

A.5.82.2 bgcolor

The **bgcolor** attribute specifies the background color of the “Footer”.

A.5.82.3 framewidth

The **framewidth** attribute specifies the frame line width of the “Footer”. When this value is “0” or not specified, no frame is rendered.

A.5.82.4 framecolor

The **framecolor** attribute specifies the frame color of the “Footer”.

A.5.82.5 framemode

The **framemode** attribute specifies the type of frame surrounding the footer area. It is possible to specify two types of frame: “square” means right angles in four corners and “curve” means rounded corners.

A.5.82.6 objid

Refer to A.5.22.19.

A.5.83 eSound

The **eSound** element specifies the information of the embedded sound.

Table A.64 – Attribute of eSound

Attribute name	Default value	Data type	Comments
objid (required)		String	Specifies the only character string in the file
refstream (required)		Object ID string	Specifies the “objid” of the SoundStream element

A.5.83.1 objid

Refer to A.5.22.19.

A.5.83.2 refstream

Refer to A.5.78.3.

A.5.84 Font

The **Font** element specifies the information of the font.

Table A.65 – Attribute of Font

Attribute name	Default value	Data type	Comments
file (required)		String	Specifies the file path and file name that contain the data
fontfilename (required)		String	Specifies the file name to be used when the font file is opened
fontname (required)		String	Specifies the font name to be used
objid (required)		String	Specifies the only character string in the file
encoding (required)		String	Specifies “TTF”, “OTF”, or “BF”

A.5.84.1 file

Refer to A.5.70.2.

A.5.84.2 fontfilename

Refer to A.5.70.1.

A.5.84.3 fontname

Refer to A.5.70.3.

A.5.84.4 objid

Refer to A.5.22.19.

A.5.84.5 encoding

Refer to A.5.70.4.

A.5.85 Image

The **Image** element specifies the information of the image. Coordinates of the attribute used by the **Image** element do not depend on various layouts, and are calculated in the coordinate system where it makes all starting points on the left.

Moreover, if the **Image** element should not be an image file that cannot be displayed by the viewer, the character string of substitution is specified for an element.

Table A.66 – Attribute of Image

Attribute name	Default value	Data type	Comments
x0 (required)	(Unsigned integer)	Decimal string	[dot]
y0 (required)	(Unsigned integer)	Decimal string	[dot]
x1 (required)	(Unsigned integer)	Decimal string	[dot]
y1 (required)	(Unsigned integer)	Decimal string	[dot]
xsize (required)	(Unsigned integer)	Decimal string	[dot]
ysize (required)	(Unsigned integer)	Decimal string	[dot]
Refstream		Object ID string	Specifies the "objid" of the ImageStream element. Only "refstream" can be selected in the Image element
objid (required)		String	Specifies the only character string in the file

A.5.85.1 x0

Refer to A.5.50.1.

A.5.85.2 y0

Refer to A.5.50.2.

A.5.85.3 x1

Refer to A.5.50.3.

A.5.85.4 y1

Refer to A.5.50.4.

A.5.85.5 xsize

Refer to A.5.50.5.

A.5.85.6 ysize

Refer to A.5.50.6.

A.5.85.7 refstream

Refer to A.5.50.7.

A.5.85.8 objid

Refer to A.5.22.19.

A.5.86 Button

The **Button** element specifies the operation when the button is operated.

There are 4 states in the button operation. Base state → Focusin state → Push state → Up state. These states are changed in turn by a user operation for the terminal.

Table A.67 – Attribute of Button

Attribute name	Default value	Data type	Comments
Objid		String	Specifies the only character string in the file

Regarding the **objid** attribute, refer to A.5.22.19

A.6 Reserved elements

The following elements may be used for the purpose described in this clause and need consensus in Interchange Parties. Otherwise these elements shall not be used.

A.6.1 ShowOrder

The **ShowOrder** element specifies the order of showing each element arranged on the **Page** element. The layout is not changed by controlling how it is shown.

A.6.2 SetText

The **SetText** element specifies the attribute change for the text.

A.6.3 div

The **div** element specifies the attribute change for the plural **P** elements at once.

A.6.4 SendMessage

The **SendMessage** element specifies a message sent to an application.

A.6.5 SimpleTextBlock

The **SimpleTextBlock** element specifies the set-up information of "Block" and "Text" to express "Block" on the screen. SimpleTextBlock has the limitation of expression for TextBlock to be treated as simple sequence.

A.6.6 MoveObj

The **MoveObj** element specifies the element that is moved from the start point to the end point within the specified time.

A.6.7 ViewPoint

The **ViewPoint** specifies the center coordinates of the area to be enlarged by the user action.

A.6.8 MiniPage

The **MiniPage** element specifies the area to express more complex layout. In the **Page** element, the "Block"s cannot be located in the character feed direction. But in the **MiniPage** element, the "Block"s can be located in the character feed direction.

A.6.9 PageDiv

The **PageDiv** element specifies the division of the text area on the **Page** element.

A.6.10 BlockList

The **BlockList** element specifies the list of the "Block"s that specifies the position in which the character flows.

A.6.11 Text

The **Text** element specifies the set-up information of "Text" to express "Block" on the screen.

A.6.12 SimpleText

The **SimpleText** element specifies the set-up information of "Text" to express "Block" on the screen.

A.6.13 Locate

The **Locate** element specifies the relative location of "Block"s in the **MiniPage** element.

A.6.14 Import

The **Import** element specifies the referred external objects and files.

A.6.15 ObjLink

The **ObjLink** element specifies the accessed external object. The format of the **ObjLink** element depends on the **acesstype** attribute.

A.6.16 FileLink

The **FileLink** element specifies the accessed external files. The format of the **FileLink** element depends on the **acesstype** attribute.

A.6.17 Solo

The **Solo** element specifies the independent “Page” of the **Main** element. The independent “Page” can only be used by the “JumpTo”, etc. action from an element in the **Main** element.

A.6.18 Pages

The **Pages** element specifies the continuous page group. The continuity of turning page is guaranteed within the same “Pages”, but there is no continuity between other “Pages”.

A.6.19 Template

The **Template** element specifies the information used as standard styles and objects when creating contents.

A.6.20 TemplateSet

The **TemplateSet** element specifies the set including the **Style** and **Objects** elements. A local “objid” is used in it.

A.6.21 PlotText

The **PlotText** element specifies the inline text.

A.6.22 AutoSpacing

The **AutoSpacing** element specifies the character string to equalize space.

A.6.23 Span

The **Span** element specifies the particular portion of the text where the text attributes are needed to change.

A.6.24 Property

The **Property** element specifies information required by authoring tools, etc. This is a reserved element at present.

Annex B (normative)

X MDF XML format

B.1 General

This annex specifies the X MDF XML format mentioned in Clause 6.

X MDF XML format is a generic format for 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 the present document, such contents will be referred to as flowing content, as opposed to paged content.

B.2 Overview of the format's structure

Flowing contents is 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 contents, 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 B.1.

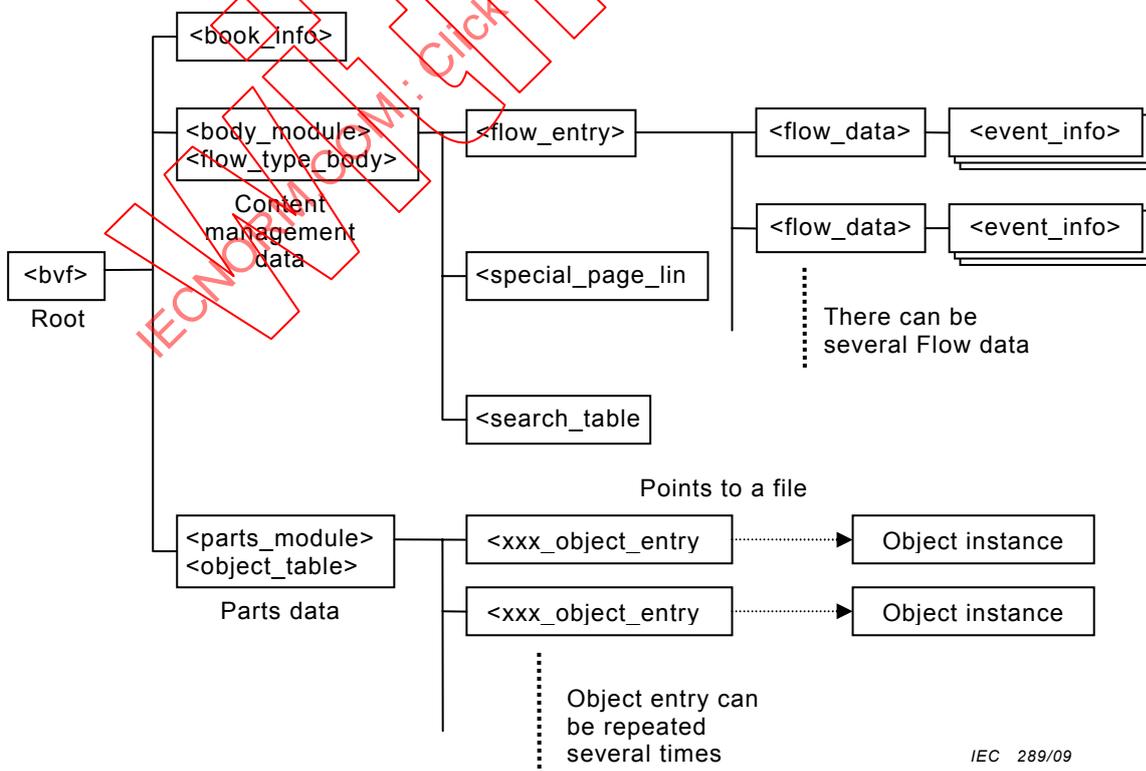


Figure B.1 – XML tree structure

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 specification is generic, and can 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 B.6. The main text will refer to it when appropriate.

B.3 Elements and attributes

B.3.1 General

The different types of values that can 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.

B.3.2 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 4 alphanumeric characters.

Example:

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

B.3.3 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 4 alphanumeric characters.

Example:

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

B.3.4 Char_ID

Char_ID specifies an identification number for character strings within text objects. It is unique in the scope of the *text object instance* (see B.4.6.2). It is a string starting by the "CR" characters, followed by 4 alphanumeric characters.

Example:

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

B.3.5 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 B.1 as a common base, while the localization B.6 will describe the language specific extensions to it.

Table B.1 – Base characters for reading

Name	Corresponding characters (All values are in Unicode)
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)

Example:

```
<title reading="PI">π</title>
```

B.3.6 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"/>
```

B.3.7 Standard character

The standard character set of the document, as set by the default_ccs attribute of the <bvf> tag (see B.4.2), 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 XML 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 B.2 and those defined for a specific localization (see B.6.2).

Table B.2 – Standard character set

Character set name	Description
"ISO/IEC 10646-IRV"	Characters in the range of US-ASCII

B.3.8 Standard character string

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

Space (0x0020) to be displayed as is.

Line feed (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.

B.3.9 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 line feed (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 extra characters should be appended to the *default_ccs* attribute of the <bvi> 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.

B.3.10 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), line feed (0x000D, 0x000A, 0x0D0A), tabulation (0x0009)) are to be handled the same way as in *Standard character strings*.

B.3.11 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 must point to the same character in all of them. 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 can 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 must match. 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 must 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 use 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 can 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 must match. Can be omitted. When this attribute is set, 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*. Can be omitted.

```
Example: <external_char alt_set="ooextchars" 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"/>
```

B.3.12 External character string

A *Standard character string* may also contain *External characters*. Unless specified otherwise, spacing characters (space (0x0020), line feed (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"/>
```

B.3.13 External extended character string

A *Standard character string* may also contain *Extended characters* and *External characters*. Unless specified otherwise, spacing characters (space (0x0020), line feed (0x000D, 0x000A), tabulation (0x0009)) should be handled the same way as they are handled in *Standard character strings*.

B.3.14 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)"/>
```

B.3.15 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 should 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*. Cannot be omitted.

Example:

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

B.3.16 Color

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

[Attributes]

color_space: Specifies the color space to 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 can be used. Acceptable color names are listed in Table B.6. 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 RBG-->
<font color="#FF0000" opacity="100"/>
<font color="black"/>
```

B.3.17 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. For details refer to <http://www.w3.org/TR/NOTE-datetime>.

Example:

```
<publication_date type="publish">1994</publication_date>
```

B.3.18 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" >
```

B.3.19 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>
```

B.3.20 Personal_name

Data format to store people's names, such as the document author. It is stored under the <personal_name> tag. Its several child elements allow do define the various parts of the name: first name, middle name and last name. This information must be entered via the tags defined below. Note that while they are all optional, you must at least write one.

[Child elements]

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

[Attribute]

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

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

[Attribute]

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

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

[Attribute]

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

Example:

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

B.3.21 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 can be specified.

[Attribute]

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

Example:

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

B.3.22 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*. Can be omitted.

<address> Stores the address as an external character string. Can be omitted.

<telephone> Stores the phone number as a *Standard character string*. Can be omitted.

<fax> Stores the fax number as a *Standard character string*. Can be omitted.

<mail_address> Stores the mail address as a *Standard character string*. Can be omitted.

- <website> Stores the home page's URI as a *Standard character string*. Can be omitted.
- <address_other_info> Allows storing additional information not covered by the preceding elements as an *External character string*. Can be omitted. If this information is to be displayed, spaces (0x0020) and line feeds (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@XXXXX.fr</mail_address>
  <website>http://www.XXXXX.fr</website>
</address_info>
```

B.3.23 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> Define whether printing is permitted or not. When omitted, it is to be considered as if the permission attribute was set to "no".

[Attribute]

permission: Define 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: Define 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>
```

B.3.24 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> tag 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*. Can be omitted.

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

[Child element]

External character string: Records the actual keyword.

Example:

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

B.3.25 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 B.3. The number of characters must be between 1 and 64, inclusive.

If a function defined by the phone number can not 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 B.3.

When the phone number string contains a function which can not be executed.

Table B.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 (1second). If a 1 second pause can not 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.)

B.3.26 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 B.4. The maximum length for local-part@domain is 256 bytes. Within the following characters, "&" (0x26) must be written as an xml entity: "&". Such xml entities are counted as one byte.

Table B.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

B.4 Description format details

B.4.1 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 B.6.8 for further details.

B.4.2 Book information modules <bvf>

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*. Can 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*. Can be omitted.
- default_ccs:** Sets the name of the character set of the *Standard characters* and *Extended characters* used in this document (see B.3). When more than one character sets are specified, they are separated by a "," (0x002C). Cannot be omitted.
- display_size:** Specifies the display (screen) size that was assumed while creating the contents, written in the standard *coordinates* format. Can be omitted. If the document includes comic flows, this size also stands for the scene size.

[Child elements]

- <book_info>** Records bibliographic data. Cannot be omitted. See B.4.2.1 for details.
- <body_module>** Content management module. Cannot be omitted. See B.4.3 for details.
- <parts_module>** Parts data modules. Cannot be omitted. See B.4.5 for details.

Example:

```
<bvf id_type="ISBN" id="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>
```

B.4.2.1 Bibliographic data <book_info>

This is where bibliographic 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. Cannot 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 can be omitted. The following attribute can be used.

[Attribute]

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

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

[Attribute]

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

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

[Attribute]

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

<edition_info> Information concerning the revision history of the book is recorded as an *External character string* in this element. Can be omitted. If the information is to be displayed, the space character (0x0020), and the line feed 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 can be used.

[Attribute]

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

<title_other_info> Other information related to the title may be stored as an *External character string* in this element. Can be omitted. If the information is to be displayed, the space character (0x0020), and the line feed 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.

<author_info> Stores the information related to the author. Can 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. Can be omitted.

<author_other_info> Other information related to the author may be stored as an *External character string* in this element. Can be omitted. If the information is to be displayed, the space character (0x0020), and the line feed 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. Can be omitted. The <publisher_info> tag has the following child elements. If <publisher_info> is specified, at least one of <publisher> or <publisher_office> must be specified as well.

[Child elements]

<publisher> Stores the information about the publisher if it is an individual. Can 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. Cannot be omitted. The following attribute can be used.

[Attribute]

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

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

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

[Attribute]

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

[Child elements]

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

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

<publisher_other_info> Other information related to the publisher may be stored as an *External character string* in this element. Can be omitted. If the information is to be displayed, the space character (0x0020), and the line feed 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.

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

[Child elements]

<seller> Stores the information about the seller if it is an individual. Can 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. Cannot be omitted. The following attribute can be used.

[Attribute]

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

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

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

[Attribute]

seller_code: Records the sellers' ID. Can be omitted.

[Child elements]

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

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

<seller_other_info> Other information related to the seller may be stored as an *External character string* in this element. Can be omitted. If the information is to be displayed, the space character (0x0020), and the line feed 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. Can be omitted. It has the following child element.

[Child element]

<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 must be at least one <book_id>. The following attribute must be set.

[Attribute]

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

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

[Child element]

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

[Attribute]

type: type of the classification used. Cannot be omitted.

<rating> Allows to rate the contents as violent, or adult. Can be omitted. The following attributes can 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. Can be omitted.

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

[Child element]

<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 must be at least one <publication_date>. The following attribute can be used.

[Attribute]

type: Specifies what type of date it is. If omitted, it will default to "publish".

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

[Child element]

<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 must be at least one <net_price>. The following attributes can 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*. Cannot be omitted.

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

<book_abstract> An abstract of the book, written as an *External character string*, is recorded in this element. Can be omitted. If the information is to be displayed, the space character (0x0020), and the line feed 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. Can be omitted. The file type as defined by it must be checked against the type attribute before opening the file. The following attribute must be used.

[Attribute]

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

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

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

<other_book_info> Other information related to the book may be stored as an *External character string* in this element. Can be omitted. If the information is to be displayed, the space character (0x0020), and the line feed 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, Fooobar city, Japan</address>
      </address_info>
    </publisher_office>
  </publisher_info>
  <book_id_info>
    <book_id type="ISBN">4-1234-5678-9</book_id>
    <book_id type="Japaneze_ID_number">454745-7</book_id>
  </book_id_info>
  <classification_info>
    <classification type="Japaneze_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>

```

B.4.3 Content management module <body_module>

B.4.3.1 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 element]

<flow_type_body> Handles the flowing contents' data. See B.4.3.2 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>

```

B.4.3.2 Flowing content data <flow_type_body>

B.4.3.2.1 <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 must only be one instance of this element. Cannot be omitted. See B.4.3.2.2 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. Can be omitted. Written as described in B.4.3.2.5. Omission of this element means there is no special page.

<search_table> Records the data needed to create a search table. Can be omitted. Written as described in B.4.3.2.6. Omission of this element means there is no search table.

B.4.3.2.2 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. Can be omitted. See B.4.3.2.3 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 must be one or more instance of this tag. The order in which flow data are recorded determines the display order. See B.4.3.2.4 for details.

B.4.3.2.3 Flow data default attribute module <flow_default_attribute>

Sets the default values of attributes that will be used to display each flow data (see B.4.3.2.4), 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 a comic object as main data.

Part of the values that can be set in this tag can 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 B.4.6.2) 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> 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. Can 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*, see B.4.6.2.2 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. Can 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 can not 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 can not handle this screen orientation, this is not necessarily applied.

[Child elements]

<flow_default_size> Defines the default letter spacing, line pitch and margin size for all the flows of the content data. Can 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. Can 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. Can 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.

margin: Default margin size. The following values are allowed. Can 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 it depends 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. Can be omitted. 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 B.4.6.2.2), 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 can be used.

[Attributes]

fontname: Default font name. More than one font can 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. Can 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. Can be omitted. If it is, the size is unspecified and depends on the viewer.

"maximum"

"big"

"medium"

"small"

"minimum"

The actual sizes of these 3 possibilities are viewer-dependent, as it depends on the capabilities of the underlying device.

bold_flag: Decides if the content data should be displayed as 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 must be displayed in bold style, except, in a *text object instance*, those within a **** tag with its **bold_flag** attribute set to "no" (see B.4.6.2.3).

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. The following values can be used.

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

"yes_only": Ruby must 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 must not be displayed, and cannot be turned on by the user. However, this does not apply if the viewer is incapable of disabling ruby display.

<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 B.4.6.2.2), 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. Can be omitted. If both this tag and **<line_breaking_method>** tags in the *text object instance* (defined in B.4.6.2.2) 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: Choose 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 B.6.

Other attributes may be added by localized methods. See B.6.

[Child elements]

None, unless added by localized methods. See B.6.

B.4.3.2.4 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*. Can be omitted. Within **<flow_entry>**, there must 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, of a "comic object", or of a "search page" object. Cannot 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 must 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 B.4.6.2.3) must be used.

[Child element]

<event_info> Event information module. Defines events (triggers) and associated actions. Can be omitted. See B.4.4 for details. If the *Object_ID* set in the *body_id* field refers to a comic object, this event_info is to be ignored.

B.4.3.2.5 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 element]

<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*. Can be omitted.

[Child element]

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

– Specify only the flow data

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

Example:

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

- 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 object within that flow. Char ID is the ID number of a string defined within that text 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>
```

B.4.3.2.6 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. From here, the entry word points to the search's target. The entry words are registered in the <key_entry> tag of the *text object instance* (see B.4.6.2.3), then 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 having verified the entry word strings. In the distribution format, entry words are stored in this search table. The <search_table> tag has the following child elements.

[Child element]

<search_table_def> Specifies information concerning each search table. There must 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 must be unique to each search table within the content and stored as a *Standard character string*. Cannot 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 B.4.6.6 for details) or from the entry word information stored in *text object instance's* <key_entry> tag (see B.4.6.2.3 for details).
- use_default: Define whether a search based on this search table can be conducted by the viewer, when the search is not

initiated from a related search page. The following values can be used. Defaults to "no" when omitted.

"yes"

"no"

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

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

Other sorting methods are defined in the localization (see B.6.5).

[Child elements]

<enable_key_type> Defines the type of characters that can 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. Cannot 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.

Please refer to B.6.6 on localization for language-specific attributes. The preceding character sets are defined in Table B.5.

Table B.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. Cannot be omitted. The interpretation of this tag differs depending on the modes of the search, viz. "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 B.4.6.6 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 should be 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 B.6.7) 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>

```

B.4.4 Event info module <event_info>

B.4.4.1 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 element]

<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 B.4.4.2 for details. There can be more than one instances of this <event> tag within <event_info>. If no event is to be specified, the <event_info> tag itself should 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

B.4.4.2 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 B.4.4.3 for details. There must be only one <trigger> tag, and it can have only one child element. Cannot be omitted.
- <action> See B.4.4.4 for details. There must be only one <action> tag, and it can have only one child element. Cannot be omitted.

Example:

```

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

```

Trigger

Action

B.4.4.3 Trigger information

The <trigger> tag has the following child elements.

B.4.4.3.1 Trigger (pointer) <trigger_pointer>

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. Cannot be omitted. Written as *Object_ID/Char_ID*.

Example:

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

The char ID part of this string may only refer to the following (see B.4.6.2.3 for details):

- a) The id number defined in the char_id attribute of an <char_id> tag.
- b) 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 B.4.6.2.3) cannot 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 element]

<pointer_region> When the trigger area must be restricted to only a part of the image pointed to by the id attribute of the <trigger_pointer> tag, this <pointer_region> tag allows to describe a *Polygonal_region* (see B.3). If what the id attribute of the <trigger_pointer> points to is not an image, the content of this element is ignored. Can 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 must trigger the first of the two events, resulting in a page jump to PG0043.

B.4.4.4 Action information

B.4.4.4.1 <action> tag

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

B.4.4.4.2 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. Cannot be omitted.

If the object is registered in the object management table (see B.4.5) (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 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 must be an id number registered in the char_id attribute of an <object> tag (B.4.6.2.3).

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>, the playback must be stopped. In case of an animation, the playback stops when it goes out of the screen.

Example:

```
<action_play object_id="OBkj23"/>
<action_play object_id="OB1234/CR0001"/>
<action_play object_id="OB1234/CR0001" action="normal"/>
```

B.4.4.4.3 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. Must be omitted if the book is omitted too, cannot be omitted when the book is not omitted.

a) When the book attribute points to a web site.

The attribute must contain: "text/html"

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

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

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

The attribute must 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 cannot 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 object (see B.4.3.2.4). In that case, the position within that text is specified using the *char_id* of the target. Can 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"/>
```

B.4.5 Parts data module <parts_module>

B.4.5.1 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 element]

<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 sub-sections. Cannot 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 |
| <comic_object_entry> | Records comic 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.

B.4.5.2 Dynamic text object <dynamic_text_object_entry>

This 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*. Cannot be omitted. Before opening the file specified in this attribute, it must 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". Cannot 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. Cannot be omitted.

[Child element]

<permission_info> Defines the permissions concerning the object pointed by the src attribute. Written in the standard *Permission* format. Can 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>

```

B.4.5.3 Sound object <sound_object_entry>

This 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. Cannot 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. Cannot be omitted. Before opening the file specified in this attribute, it must be checked against the type attribute. Currently, only mp3 (extension: .mp3), AAC (extension: .3g2) and SMAF (extension: .mmf) are allowed.

[Child element]

<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 *object_ID* format. Cannot be omitted.

Example:

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

B.4.5.4 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. Cannot be omitted. Before opening the file specified in this attribute, it must 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. Cannot 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. Cannot be omitted.

[Child element]

<permission_info> Defines the permissions concerning the object pointed by the *src* attribute. Written in the standard *Permission* format. Can 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>
```

B.4.5.5 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. Cannot be omitted. Before opening the file specified in this attribute, it must 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. Cannot 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. Cannot be omitted.

Example:

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

B.4.5.6 Comic object <comic_object_entry>

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

[Attributes]

- src: Filename of the comic object, written in the standard *Filename* format. Cannot be omitted. Before opening the file specified in this attribute, it must be checked against the type attribute.
- type: Records the MIME type of the comic object. In the current standard, only "text/x-bvf-comic" is permitted. Cannot 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 *object_ID* format. Cannot be omitted.

[Child element]

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

Example:

```
<object_table>
...
  <comic_object_entry src="comic1.xml" type="application/x-bvf-comic"
object_id="OB0001">
    <permission_info>
      <print_permission permission="no"/>
    </permission_info>
  </comic_object_entry>
...
</object_table>
```

B.4.6 Object instances

B.4.6.1 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

Image object instance

Sound object instance

Animation object instance

Search page object instance

Movie object instance

Comic object instance

B.4.6.2 Text object instance <text_data>

B.4.6.2.1 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>).

B.4.6.2.2 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 B.4.3.2.3, 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 B.4.3.2.4). 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 B.2.

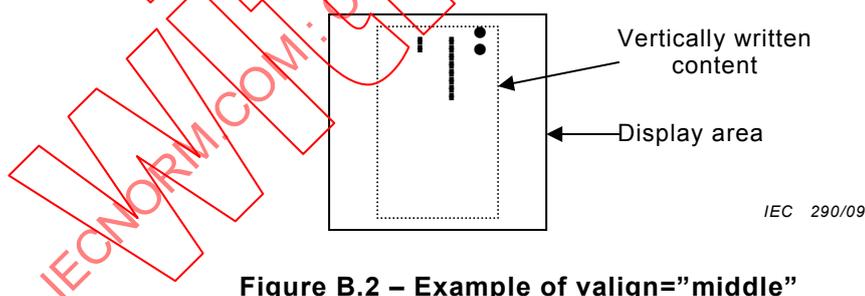


Figure B.2 – Example of valign="middle"

[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 attribute.

[Attribute]

- fontname: Default font name. More than one font can 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. It has the following attribute.

color_space, color, opacity:	Defines the font color to be used for the content data. Written in the standard <i>color</i> 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 can be used. <ul style="list-style-type: none"> “yes”: “yes_only”: “no”: “no_only”:

<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.

[Attribute]

color_space, color, opacity:	Defines the color to be used as background color. Written in the standard <i>color</i> data format. When omitted, defaults to the <flow_default_attribute> value.
type:	Stores the MIME type of the background image. Can be omitted only if the src attribute is also omitted. The possible values are listed below. <ul style="list-style-type: none"> "image/png" "image/jpeg" "image/x-ccf"
src:	Sets the filename of the image to use as a background image, written in the standard <i>Filename</i> format. Can be omitted. Before opening the file specified in this attribute, it must 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 <i>Permission</i> format.
--------------------	---

Can 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.

[Attribute]

- type: Stores the MIME type of the background music. Cannot 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 *Filename* format. Cannot be omitted. Before opening the file specified in this attribute, it must 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. Can 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: Choose 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 B.6.

Other attributes may be added by localized methods. See B.6.

[Child elements]

None, unless added by localized methods. See B.6.

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 >
```

B.4.6.2.3 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 would start at the beginning of a line even without inserting a line break.

If what follows the paragraph would start 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 starts the new line after the image, <br clear="all"/> must 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 must be "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 can 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 percentage of the line length (column in case the writing direction is vertical). When omitted, defaults to "0em". Negative values can not be used. Paragraphs may be contained in another paragraph. In that case, the margins add. 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 percentage of the total line size (column in case the writing direction is vertical). When omitted, defaults to "0em". Negative values cannot be used. Paragraphs may be contained in another paragraph. In that case, the margins add. Percentages are

also relative to the inner area thus calculated, not the total display area.

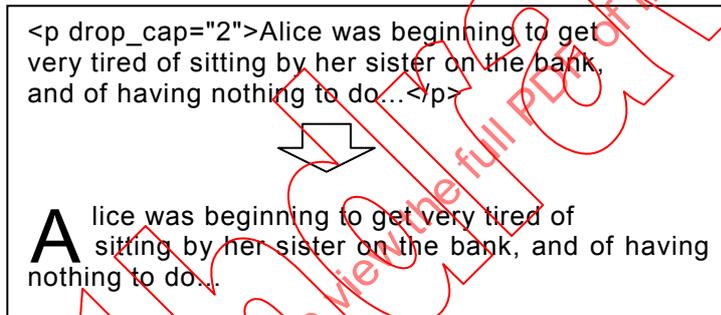
align: Determines whether the text should be in the center, left-aligned or right-aligned within the line size defined by the top, bottom and first line indent attributes. This attribute accepts the values listed below. When omitted, the current setting is kept unchanged.

"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 B.3.



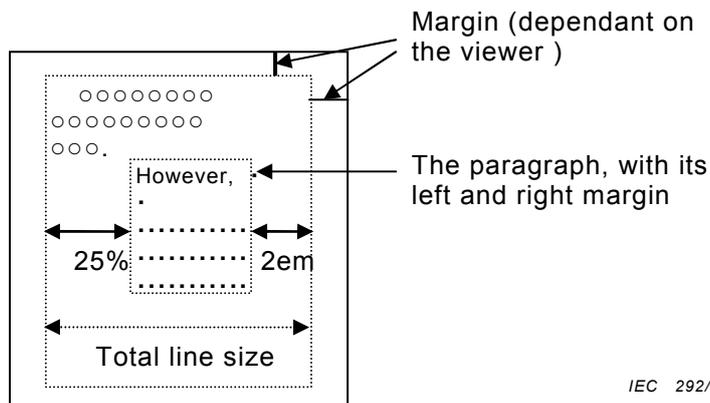
IEC 291/09

Figure B.3 – Example of dropped capital

[Child element]

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

Figure B.4 shows an example of a horizontally written paragraph with top="25%" and bottom="2em".



IEC 292/09

Figure B.4 – 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 will be 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>` 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 can 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 cannot 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 must 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 can 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 characters' color. Written in the standard *Color* data format (see.B.3.16 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 possible values are "yes" and "no". When omitted, the current state is kept.
- underline: Defines whether the string included in the tag should be underlined or not. However, objects included inside the tag should not be underlined. The possible values are "yes" and "no". When omitted, the current state is kept.

[Child elements]

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

Example:

```

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

<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 B.5. When the text is written horizontally, this tag has no effect. The following child element is accepted.

[Child element]

Extended external character strings can be used here.

Example:

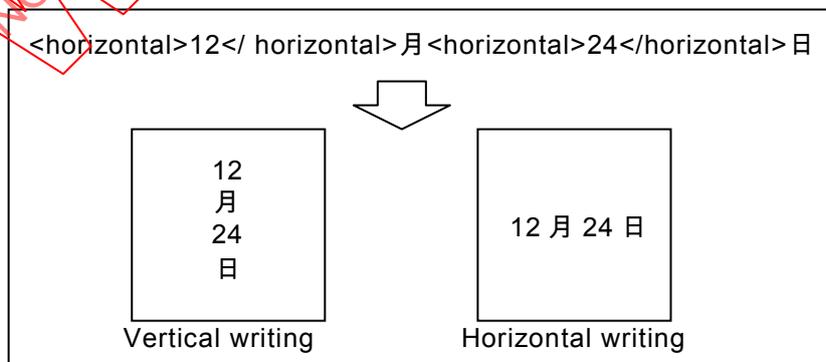


Figure B.5 – Horizontal writing in vertical text

<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 B.6. The <ruby> tag has the following child elements.

[Child elements]

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

Example:

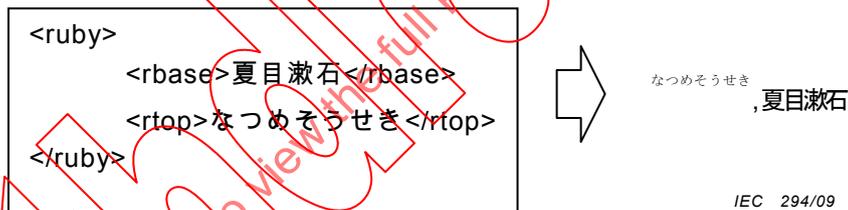


Figure B.6 – Ruby

<external_char> Inserts an *External character*. The detailed syntax is described in B.3.11. Within the *text object instance*, the alt_img and alt_vimg attributes can 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 char_id 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 must be in the same *text object instance*. Note that the char id used here cannot 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. Can be omitted.

mask_type: Selects the type / style of mask to be used. The values listed below can 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 B.3.16. Must be defined only if the mask_type attribute is set to "color". If omitted, defaults to "black".

hold_flag: Define the state recording policy, allowing remembering whether the mask has been turned on or off. Can be set to the values listed below. Defaults to "scope" when omitted.

"scope": Remember the state as long as the current *text object instance* is displayed.

"on_power": Remember 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 can be any combination of *External character strings* and the
, , <horizontal>, <ruby> and <object> tags. However, the following restrictions apply to the <object> tag:

The src attribute must point to a image file.

The char_id attribute cannot 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. Cannot be omitted.

[Child elements]

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

<key_entry> </key_entry> Defines a list of keywords to be used for searching capabilities. It registers the word for inclusion in a search table, but will not be displayed in the main body of text. When the keyword has been selected by the user among the search results, what is displayed is the text following the </key_entry> closing tag. The following child element is available.

[Child element]

<key_item> Records a single keyword. There must be at least one <key_item> per <key_entry>. It has the following attributes and child elements.

[Attributes]

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 B.4.3.2.6). Cannot be omitted.

search_word: Defines the actual keyword. Only the characters allowed in the corresponding search table can be used. See <enable_key_type> for details in B.4.3.2.6. Cannot be omitted.

[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*.

Example:

```
<key_entry>
  <key_item search_word="color" table_id="ST0001">
    Color
  </key_item>
  <key_item search_word="colour" table_id="ST0002">
    Colour
  </key_item>
</key_entry>
<!-- What follows is what will be displayed
      as the result of the search -->
A color (English spelling: colour) is ...
```

`<page_break/>` Inserts a page break. This tag has attributes listed below. It cannot be used as a child element of any of the tags that can 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 cannot 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 B.4.3.2.4).

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. Cannot be omitted. In the current version of this standard, the following types are permitted:

Png image: "image/png"

Jpeg image: "image/jpeg"

Animation object: "application/x-bvf-flip-animation"
(see B.4.6.5)

`src`: Sets the filename of the object to be inserted, written in the standard *Filename* format. Cannot be omitted. It must 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*. Can 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.



"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.



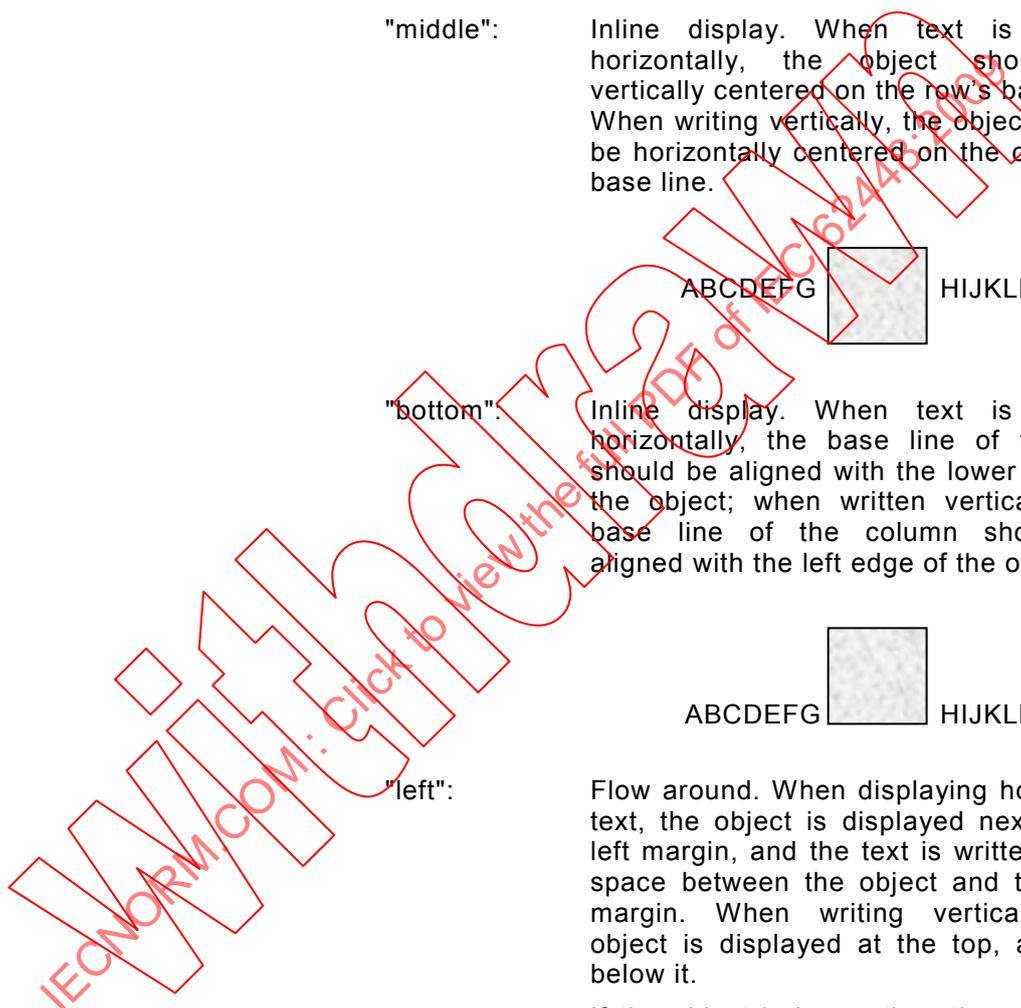
"bottom": Inline display. When text is written horizontally, the base line of the row should be aligned with the lower edge of the object; when written vertically, the base line of the column should be aligned with the left edge of the object.



"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 can 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".

ABCDEFGHIJKL



"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 can 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



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 cannot be used by the <action_play> (see B.4.4.4.2) 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 B.4.4.2).

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 was set to "1".

[Child element]

`<permission_info>` Defines the permissions about the object referred to by the `src` attribute. Written in the standard *Permission* format. Can 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"/>
    
```

B.4.6.3 Image object instance

Regular image file referred to by the `<object>` tag of the *text object instance* (see B.4.6.2.3). In the present state of this standard, it can either be a PNG file or a JPEG file.

B.4.6.4 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 can be used.

B.4.6.5 Animation object instance

The following possibilities are available.

B.4.6.5.1 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 "1 s" 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. Can be omitted. It has the following attributes.

[Attributes]

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

src: Name of the sound file, written in the standard *Filename* format. Cannot be omitted. Before opening the file specified in this attribute, it must 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 must be at least one instance of this tag. When there are more than one of this tag in the definition of an animation object, all the images referred to by the src attribute must 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. Cannot 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. Cannot be omitted. Before opening the file specified in this attribute, it must 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 must 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>
```

B.4.6.6 Search page object instance <search_page>

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

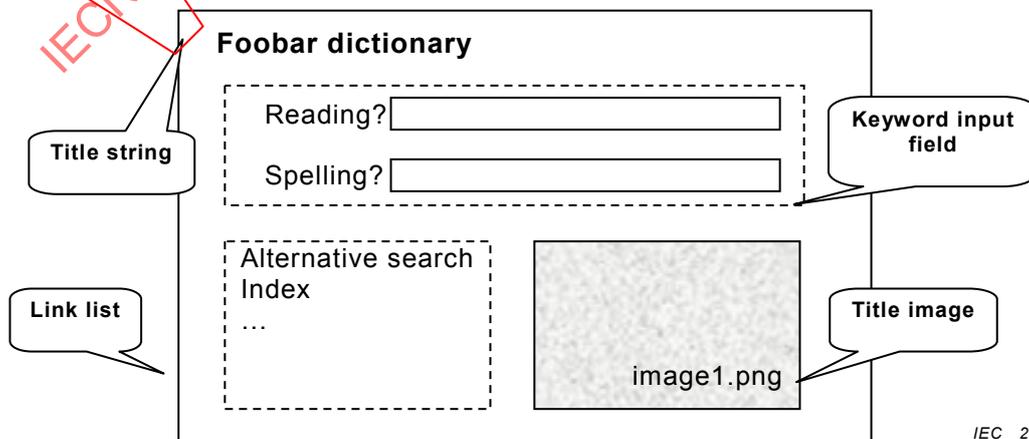
Example:

```
<?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 B.7. 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.



IEC 295/09

Figure B.7 – 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. Can 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. Can 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. Can be omitted. Before opening the file specified in this attribute, it must be checked against the type attribute.

[Child element]

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

Example:

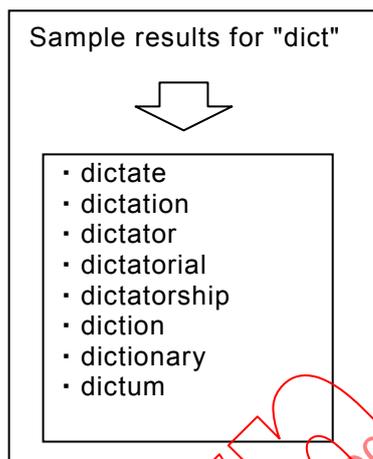
```

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

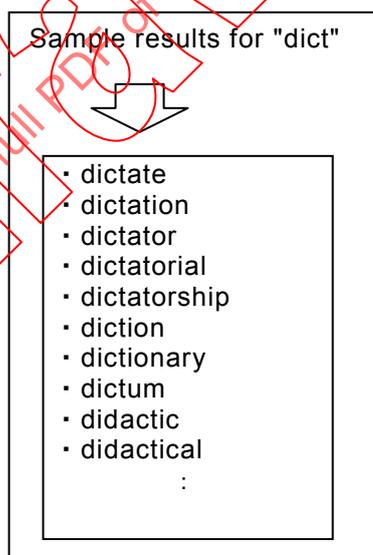
`<key_input_region>` Defines the keyword input region. Within a single search screen, there can only be one or two instances of this element. The viewer uses this information to display the input region, and then conducts the search based on user input. It has the following attributes and child elements.

[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 B.4.3.2.6). Cannot be omitted.
- `search_type`: Defines the type of search, to be chosen from the possibilities listed below. Can be omitted, and defaults to "matches_only" in that case.
- "matches_only": "Matches only" search mode. The normalized (see B.4.3.2.6, `<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.



"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.



[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 B.7. Cannot be omitted.

[Child element]

The character string to be used as explained above.

<enable_key_type> Defines what kind of characters 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 B.4.3.2.6), so the details will not be repeated here. However, the character types allowed in the search page object's <enable_key_type> must be a subset of those defined in the associated search table. Cannot 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 B.4.4.3.1) and <action_page_jump> (see B.4.4.4.3) tags. They should be recorded in the event data (see B.4.4) of the flow hosting this search page object. Cannot 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. Cannot be omitted.

[Child element]

<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 *Extended external character string*. Cannot be omitted.

Example:

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

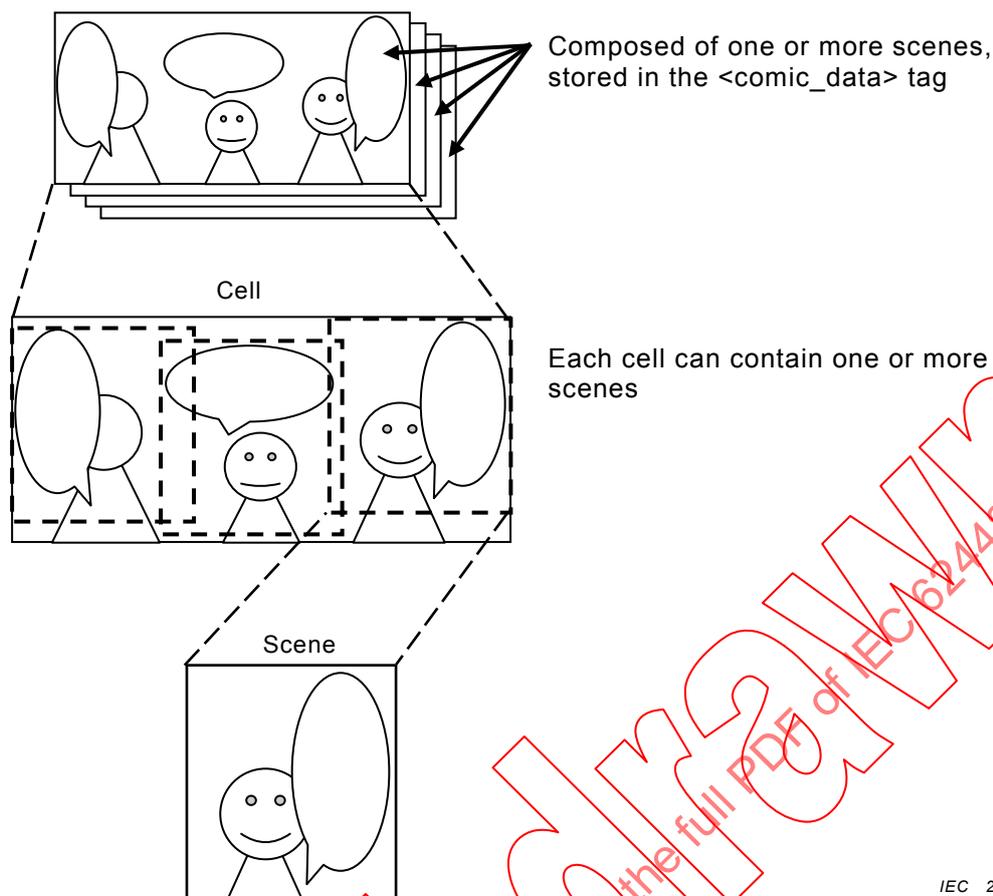
<search_link_item char_id="CR0002">
  <search_link_title>Index</search_link_title>
</search_link_item>
```

B.4.6.7 Movie object instance

Regular movie file referred to by the <movie_object_entry> tag (see B.4.5.5). In the present state of this standard, only 3GPP2 files are allowed.

B.4.6.8 Comic object instance <comic_data>

Registers a comic object instance, whose structure is depicted in Figure B.8. As can be seen, the comic object instance is composed of one or more cells, which in turn can have one or more scenes.



IEC 296/09

Figure B.8 – Comic object Instance

The viewer is to display the comic object in the following fashion:

- a) display the first cell, at the position recorded as the first scene;
- b) wait for user to press the key responsible for scrolling;
- c) scroll to the second scene in the first cell;
- d) wait for user to press the key for scrolling;
- ...
- e) display the second cell, at the position recorded as the first scene;
- ...

The viewer should display the next cell only after all the scenes in the current one have been shown, waiting for the user input between each one.

How to divide data into comic object instances is up to the data preparer. For instance, each chapter or story of the comic book may be represented as a separate comic object instance, or they may be all grouped into one.

The <comic_data> tag has the following child element.

[Child elements]

<comic_body> Stores the content data of the comic object instance. Cannot be omitted. There can be only one <comic_body> tag under <comic_data>. See B.4.6.8.1 for details.

B.4.6.8.1 Comic content data <comic_body>

This field stores the main content data, that is to say the data that will actually be displayed. In the present standard, only the cell/scene structure presented in B.4.6.8 is available.

[Child elements]

<comic_cell_type_body> Serves as a main container for cell/scene structured data. Cannot be omitted. There must be only one instance of this tag. It has the following child element and attributes.

[Attributes]

bg_color, bg_color_space, bg_opacity:

Defines the default background color of this cell/scene structure content, in the standard *color* format. This setting may be overridden for any particular cell by the values defined in the corresponding attributes of the <cell> tag (see B.4.6.8.2). Can be omitted. If both these attributes and those of the <cell> tag are omitted, the background color defaults to the values set in <flow_default_background> (see B.4.3.2.3). If none is defined, the behavior depends on the viewer, and may follow user defined preferences.

[Child element]

<cell>

Stores information for each individual cell. There must be at least one instance of this tag. See B.4.6.8.2 for detailed explanations.

B.4.6.8.2 Cell data <cell>

Stores information defining an individual cell. It has the following attributes and child elements.

[Attributes]

src: Specifies the image to use as a base for this cell. Written in the *filename* standard file format. In the current specification, JPEG, PNG and CCF files are allowed. Cannot be omitted. Before opening the file specified in this attribute, it must be checked against the type attribute. The image specified by this attribute will be displayed by the viewer according to the other attributes and child elements, such as scroll, fit, effect, or <cell_scene>.

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

“image/png”

“image/jpeg”

“image/x-ccf”

char_id: Defines the ID number of this cell, written in the standard *char ID* format. Can be omitted. It is used as a target for jumps from the viewer’s menu or from link jumps in text object instances.

bg_color, bg_color_space, bg_opacity:

Defines the default background color of this cell, in the standard *color* format. This setting overrides any default cell setting defined in the corresponding attributes of the <comic_cell_type_body> tag (see B.4.6.8.1). Can be omitted. If both these attributes and those of the <comic_cell_type_body> tag are omitted, the background color defaults to the values set in <flow_default_background> (see B.4.3.2.3). If none is defined, the behavior depends on the viewer, and may follow user defined preferences.

- fit:** When the actual display size is smaller than the one specified by the “display_size” attribute of the <bvf> tag (See B.4.2), this attribute specifies whether or not the viewer is allowed to resize the image (preserving the aspect ratio) to fit into the screen. The possible values are “yes” and “no”, and it defaults to “no” when omitted. Note that if the cell contains one or more scenes for which scroll is defined (the scroll_time attribute being set to a non-zero value), scrolling takes priority over resizing, and this attribute is treated as if it were set to “no”.
- effect:** Defines an animation effect to use when first displaying the image of this cell, either right after opening the document, or when this cell is displayed right after another. If omitted, no effect should be used. The following values can be used:
- “fade-in”: the screen is set to the background color, and the new cell’s image is displayed progressively fading in.
 - “dissolve”: the current cell’s image is displayed progressively over the previous cell’s image. This is similar to the fade-in effect, except that the screen is not set to the background color before displaying the new cell.
 - “slide-right”: the screen is set to the background color, and the new cell’s image slides in rightwards, starting from left.
 - “slide-left”: the screen is set to the background color, and the new cell’s image slides in leftwards, starting from right.
 - “slide-up”: the screen is set to the background color, and the new cell’s image slides in upwards, starting from bottom.
 - “slide-down”: the screen is set to the background color, and the new cell’s image slides in downwards, starting from top.
 - “overwrite-right”: the new cell’s image slides in rightwards, starting from left, covering the previous cell’s image.
 - “overwrite-left”: the new cell’s image slides in leftwards, starting from right, covering the previous cell’s image.
 - “overwrite-up”: the new cell’s image slides in upwards, starting from bottom, covering the previous cell’s image.
 - “overwrite-down”: the new cell’s image slides in downwards, starting from top, covering the previous cell’s image.
 - “pushout-right”: the new cell’s image slides in rightwards, starting from left, while the previous cell’s image slides out of the screen in the same direction.
 - “pushout-left”: the new cell’s image slides in leftwards, starting from right, while the previous cell’s image slides out of the screen in the same direction.
 - “pushout-up”: the new cell’s image slides in upwards, starting from bottom, while the previous cell’s image slides out of the screen in the same direction.
 - “pushout-down”: the new cell’s image slides in downwards, starting from top, while the previous cell’s image slides out of the screen in the same direction.
 - “box-center”: the screen is set to the background color, and a progressively enlarging square portion of the new cell’s image is displayed, starting in the center, and growing outwards.
 - “box-rightdown”: the screen is set to the background color, and a progressively enlarging square portion of the new cell’s image is displayed, starting at the upper left corner, and growing towards lower right.

- “box-leftdown”: the screen is set to the background color, and a progressively enlarging square portion of the new cell’s image is displayed, starting at the upper right corner, and growing towards lower left.
- “box-rightup”: the screen is set to the background color, and a progressively enlarging square portion of the new cell’s image is displayed, starting at the lower left corner, and growing towards upper right.
- “box-leftup”: the screen is set to the background color, and a progressively enlarging square portion of the new cell’s image is displayed, starting at the lower right corner, and growing towards upper left.
- “box-free”: the screen is set to the background color, and a progressively enlarging square portion of the new cell’s image is displayed, starting at the position specified by the effect_pos attribute, and growing outwards.
- effect_time: determines the speed of the effect animation, on a 1 to 10 scale, 1 being the fastest, and 10 the slowest. Note that the actual speed depends on the viewer and the device in use. Defaults to 5 when omitted.
- effect_pos: when the effect is set box-free, this attributes defines the position within the scene where the square will grow from. Cannot be omitted when effect is set to box-free, and is ignored if the attribute is set otherwise or not set.
- scroll: defines the scrolling method for the cell. If omitted, defaults to “no”. The following values can be used.
- “custom” specifies that the initial display position and scrolling path is “custom-set”, using the position attribute of the <cell_scene > tag.
 - “up” displays the image from its lower end, and scrolls to the upper end after user input.
 - “down” displays the image from its upper end, and scrolls to the lower end after user input.
 - “left” displays the image from its rightmost end, and scrolls to the leftmost end after user input.
 - “right” displays the image from its leftmost end, and scrolls to the rightmost end after user input.
 - “no” displays the image centered, without doing any scrolling.
- scroll_time: determines the speed of the scrolling animation, on a 1 to 10 scale, 1 being the fastest, and 10 the slowest. Note that the actual speed depends on the viewer and the device in use. Defaults to 5 when omitted. It can also be set to 0, in which case the viewer should jump from the initial position to the final one, instead of using a smooth scrolling animation. If the scroll attribute is set to “custom”, this attribute is ignored.
- vibration: specifies whether the viewer should make the device vibrate when displaying the cell. The possible values are “on” and “off”. If omitted, defaults to “off”. Vibration and displaying the image should start at the same time. If the scroll attribute is set to “custom”, this attribute is ignored.
- vibration_time: if the vibration attribute is set to “on”, this attribute defines how long the vibration should last, on a 5 step scale, 1 being the shortest, and 5 the longest. Actual duration depends on the device on which the viewer is installed. When omitted, defaults to 3. If the scroll attribute is set to “custom”, this attribute is ignored.
- sound: designates a sound file to be played when the image is displayed. Written in the standard *filename* format. The sound file is just played once. Before opening the file specified in this attribute, it must be checked against the

sound_type attribute. When omitted, no sound is to be played. If the scroll attribute is set to “custom”, this attribute is ignored.

sound_type: Stores the MIME type of the sound file. Cannot be omitted if the sound attribute is defined. The possible values are listed below. If the scroll attribute is set to “custom”, this attribute is ignored.

- "audio/mp3"
- "audio/3gpp2"
- "application/x-smaf"

url_jump: Associates the URL of a web site with the current cell. Only URL beginning with http:// and https:// are allowed. Refer to the RFC 3986 for details about the format to use. When omitted, no web site is associated with the cell. If the scroll attribute is set to “custom”, this attribute is ignored.

[Child element]

<cell_scene> Stores information concerning each scene of the cell. Can only be used when scroll is set to “custom”. There is one instance of this tag for each scene.

Example:

```

<comic_data>
  <comic_body>
    <comic_cell_type_body bg_color="white">
      <cell src="cell001.png" type="image/png" char_id="CR0001"
bg_color="blue" effect="fade-in" scroll="custom">
        <cell_scene ... />
        <cell_scene ... />
        ...
      </cell>
      <cell src="cell002.png" type="image/png" char_id="CR0002"
scroll="custom">
        <cell_scene ... />
        <cell_scene ... />
        ...
      </cell>
      <cell src="cell003.png" type="image/png" fit="yes" scroll="no"
vibration="on"/>
      <cell src="cell004.png" type="image/png" scroll="up"
scroll_time="10" url_jump="http://www.sharp.co.jp"/>
      ...
    </comic_cell_type_body>
  </comic_body>
</comic_data>

```

B.4.6.8.3 Scene data <cell_scene>

Stores information for each individual scene. It has the following attributes.

[Attributes]

- position: Specifies one point in the cell's image which is to be considered the center of the scene, in the standard *coordinate* format. The scene should be displayed so that this point is located in the center of the screen. If omitted, it defaults to the center of the cell's image.
- scroll_time: Determines the speed of the scrolling animation from this scene to the next one, on a 1 to 10 scale, 1 being the fastest, and 10 the slowest. Note that the actual speed depends on the viewer and the device in use. Defaults to 5

- when omitted. It can also be set to 0, in which case the viewer should jump from the initial position to the final one, instead of using a smooth scrolling animation. If this scene is the last one in the cell, this attribute is ignored.
- vibration:** Specifies whether the viewer should make the device vibrate when displaying this scene. The possible values are “on” and “off”. If omitted, defaults to “off”. Vibration and displaying the scene should start at the same time.
- vibration_time:** If the vibration attribute is set to “on”, this attribute defines how long the vibration should last, on a 5 step scale, 1 being the shortest, and 5 the longest. Actual duration depends on the device on which the viewer is installed. When omitted, defaults to 3.
- sound:** Designates a sound file to be played when this scene is displayed. Written in the standard *filename* format. The sound file is just played once. Before opening the file specified in this attribute, it must be checked against the *sound_type* attribute. When omitted, no sound is to be played.
- sound_type:** Stores the MIME type of the sound file. Cannot be omitted if the sound attribute is defined. The possible values are listed below.
- "audio/mp3"
 - "audio/3gpp2"
 - "application/x-smaf"
- url_jump:** Associates the URL of a web site with the current scene. Only URL beginning with `http://` and `https://` are allowed. Refer to the RFC 3986 for details about the format to use. When omitted, no web site is associated with the cell.

Example:

```
<cell src="cell001.png" type="image/png" char_id="CR0001" bg_color="blue">
  <cell_scene position="(30,50)" scroll_time="1" sound="sound1.mp3"
  sound_type="audio/mp3"/>
  <cell_scene position="(100,200)" vibration="on" vibration_time="4"
  url_jump="http://www.sharp.co.jp"/>
  ...
</cell>
```

B.5 Available color names

The color names that can be used in the attributes specifying color types are listed in Table B.6, with the corresponding RGB values.

Table B.6 – 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	#EEE8AA	#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	

B.6 Localization

B.6.1 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.

B.6.2 Standard characters

For localization purposes, the character sets listed in Table B.7 may be used for the standard character set of the document, depending on the language.

Table B.7 – 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)

B.6.3 Characters usable for reading

In addition to the basic characters allowed in the *Reading* data format, the language-specific extensions listed in Table B.8 are proposed.

Table B.8 – Example of additional characters usable for readings

Target language	Name	Description
Japanese	Full-width Japanese katakana characters	ァアイゥウエェオオカガキグクゲケゴサザシジスズセゼソタダチッツツテトドナニヌネノハ/V/Vヒビピフ ブヘベペホボボマミムメモヤユヨヨラリルレロワヰヱヱヱヱヱヱ (0x30A1 to 0x30F6)
Japanese	Japanese long vowel mark	ー (0x30FC)
French	French alphabet extensions	À (0x00C0) Â (0x00C2) Æ (0x00C6) Ç (0x00C7) È (0x00C8) É (0x00C9) Ê (0x00CA) Ë (0x00CB) Ï (0x00CE) Î (0x00CF) Ó (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) ï (0x00EE) î (0x00EF) ô (0x00F4) ù (0x00F9) û (0x00FB) ü (0x00FC) ÿ (0x00FF) Œ (0x0152) œ (0x0153)

B.6.4 Line breaking methods

B.6.4.1 For Japanese language

Possible value for the method attribute of the <flow_default_line_breaking_method> (see B.4.3.2.3) and <line_breaking_method> (see B.4.6.2.2) 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>
```

B.6.4.2 For the English language

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.

B.6.5 Sorting rules for <search_table_def>

The methods listed in Table B.9 are proposed.

Table B.9 – 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 À Â Ã Ä Å Æ B C Ç È É Ê Ë Ì Í Î Ï J K L M N O Ô Õ Æ P Q R S T U Û Ü Ü V W X Y Ÿ 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.

B.6.6 Additional attributes for <enable_key_type>

It is proposed that <enable_key_type> have the language-specific attributes listed in Table B.10 in addition to those described in B.4.3.2.6. Each accepts 2 values; "yes" and "no". Defaults to "no" when omitted.

Table B.10 – 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ムメモヤユヨラリルロワン (0xFF71 to 0xFF9F) and ヲアイウエオヤユヨツ (0xFF66 to 0xFF6F) Full-width katakana: アアイウウエエオオカガキククケケゴゴサザシジスズセゼソゾタチチツツテテトドナニヌネノハバヰビピフブ プヘベボボボマミムメモヤユヨヨラリルレロウワヰヰヰ (0x30A1 to 0x30F6) Full-width hiragana: ああいうええおおかがきぎくぐけげごさざしじすずせぜそぞただちぢっつっつてとどなにぬねのはばびひびびふぶ ぶへべほほほまみむめもやゆよよりりるれろわわゐゑをん (0x3041 to 0x3093) Long vowels: ー (0x30FC, full-width) and ー (0xFF70, half-width)																																				
French	French alphabet extensions	<table border="0"> <tr> <td>À (0x00C0)</td> <td>Â (0x00C2)</td> <td>Æ (0x00C6)</td> <td>Ç (0x00C7)</td> </tr> <tr> <td>É (0x00C8)</td> <td>É (0x00C9)</td> <td>Ê (0x00CA)</td> <td>Ë (0x00CB)</td> </tr> <tr> <td>Î (0x00CE)</td> <td>Ï (0x00CF)</td> <td>Ô (0x00D4)</td> <td>Ö (0x00D6)</td> </tr> <tr> <td>Û (0x00D9)</td> <td>Û (0x00DB)</td> <td>Ü (0x00DC)</td> <td>ÿ (0x00DF)</td> </tr> <tr> <td>à (0x00E0)</td> <td>â (0x00E2)</td> <td>æ (0x00E6)</td> <td>ç (0x00E7)</td> </tr> <tr> <td>è (0x00E8)</td> <td>é (0x00E9)</td> <td>ê (0x00EA)</td> <td>ë (0x00EB)</td> </tr> <tr> <td>î (0x00EE)</td> <td>ï (0x00EF)</td> <td>ô (0x00F4)</td> <td>ù (0x00F9)</td> </tr> <tr> <td>û (0x00FB)</td> <td>ü (0x00FC)</td> <td>ÿ (0x00FF)</td> <td>Œ (0x0152)</td> </tr> <tr> <td>œ (0x0153)</td> <td></td> <td></td> <td></td> </tr> </table>	À (0x00C0)	Â (0x00C2)	Æ (0x00C6)	Ç (0x00C7)	É (0x00C8)	É (0x00C9)	Ê (0x00CA)	Ë (0x00CB)	Î (0x00CE)	Ï (0x00CF)	Ô (0x00D4)	Ö (0x00D6)	Û (0x00D9)	Û (0x00DB)	Ü (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)			
À (0x00C0)	Â (0x00C2)	Æ (0x00C6)	Ç (0x00C7)																																			
É (0x00C8)	É (0x00C9)	Ê (0x00CA)	Ë (0x00CB)																																			
Î (0x00CE)	Ï (0x00CF)	Ô (0x00D4)	Ö (0x00D6)																																			
Û (0x00D9)	Û (0x00DB)	Ü (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)																																						

B.6.7 Normalization methods for <key_normalization>

B.6.7.1 For Japanese language

Additional attributes of the <key_normalization> tag (see B.4.3.2.6).

- 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.

B.6.7.2 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".

B.6.8 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.

B.7 Specification of the XMDF XML format in the RELAX NG syntax

The syntax of the XMDF XML 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 {  
        personal_name_attlist,  
        ((first_name?, middle_name?, last_name?)  
         | (first_name?, last_name?, middle_name?)  
         | (last_name?, first_name?, middle_name?)  
         | (last_name?, middle_name?, first_name?))  
    }
```

personal_name_attlist &= empty

first_name = element first_name { first_name_attlist, TextWithGaiji }

first_name_attlist &= attribute reading { text }?

middle_name = element middle_name { middle_name_attlist, TextWithGaiji }

middle_name_attlist &= attribute reading { text }?

last_name = element last_name { last_name_attlist, TextWithGaiji }

last_name_attlist &= attribute reading { text }?

organization_name =

```
    element organization_name { organization_name_attlist, TextWithGaiji }
```

organization_name_attlist &= attribute reading { text }?

address_info =

```
    element address_info {  
        address_info_attlist,  
        postal_code?,  
        address?,  
        telephone?,  
        fax?,  
        mail_address?,  
        website?,  
        address_other_info?
```

```
    }
```

```
address_info_attlist &= empty
postal_code = element postal_code { postal_code_attlist, text }
postal_code_attlist &= empty
address = element address { address_attlist, TextWithGaiji }
address_attlist &= empty
telephone = element telephone { telephone_attlist, text }
telephone_attlist &= empty
fax = element fax { fax_attlist, text }
fax_attlist &= empty
mail_address = element mail_address { mail_address_attlist, text }
mail_address_attlist &= empty
website = element website { website_attlist, text }
website_attlist &= empty
address_other_info =
    element address_other_info {
        address_other_info_attlist, TextWithGaiji
    }
address_other_info_attlist &=
    [ a:defaultValue = "preserve" ]
    attribute xml:space { "default" | "preserve" }?
permission_info =
    element permission_info {
        permission_info_attlist, print_permission?, copy_permission?
    }
permission_info_attlist &= empty
print_permission =
    element print_permission { print_permission_attlist, empty }
print_permission_attlist &=
    [ a:defaultValue = "no" ]
    attribute permission { "authorized" | "no" }?
copy_permission =
```

```
    element copy_permission { copy_permission_attlist, empty }
copy_permission_attlist &=
    [ a:defaultValue = "no" ]
    attribute permission { "authorized" | "in_device_only" | "no" }?
```

```
protection = element protection { protection_attlist, empty }
protection_attlist &= empty
```

```
keyword_list = element keyword_list { keyword_list_attlist, keyword+ }
keyword_list_attlist &= empty
keyword = element keyword { keyword_attlist, TextWithGaiji }
keyword_attlist &=
    attribute category { text }?,
    attribute reading { text }?
```

```
vertex = element vertex { vertex_attlist, empty }
vertex_attlist &= attribute position { text }
```

```
bvf = element bvf { bvf_attlist, book_info, body_module, parts_module }
bvf_attlist &=
    attribute id_type { text }?,
    attribute id { text }?,
    attribute default_ccs { text },
    attribute display_size {text}?
```

```
book_info =
    element book_info {
        book_info_attlist,
        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?
    }
book_info_attlist &= empty

title_info =
    element title_info {
        title_info_attlist,
        series_title?,
        title,
        subtitle?,
        edition_info?,
        title_other_info?
    }
title_info_attlist &= empty

series_title =
    element series_title { series_title_attlist, TextWithGaiji }
series_title_attlist &= attribute reading { text }?

title = element title { title_attlist, TextWithGaiji }
title_attlist &= attribute reading { text }?

subtitle = element subtitle { subtitle_attlist, TextWithGaiji }
subtitle_attlist &= attribute reading { text }?

edition_info =
```

```
    element edition_info { edition_info_attlist, TextWithGaiji }
edition_info_attlist &=
    attribute this_version { text }?,
    [ a:defaultValue = "preserve" ]
    attribute xml:space { "default" | "preserve" }?
title_other_info =
    element title_other_info { title_other_info_attlist, TextWithGaiji }
title_other_info_attlist &=
    [ a:defaultValue = "preserve" ]
    attribute xml:space { "default" | "preserve" }?

author_info = element author_info { author_info_attlist, author+ }
author_info_attlist &= empty
author =
    element author {
        author_attlist,
        (personal_name | organization_name),
        address_info?,
        author_other_info?
    }
author_attlist &=
    [ a:defaultValue = "author" ]
    attribute role {
        "author"
        | "editor"
        | "translator"
        | "supervisor"
        | "designer"
        | "photographer"
        | "illustrator"
        | "binder"
        | "planner"
```

```
| "other"
}?)
author_other_info =
    element author_other_info { author_other_info_attlist, TextWithGaiji }
author_other_info_attlist &=
    [ a:defaultValue = "preserve" ]
    attribute xml:space { "default" | "preserve" }?

publisher_info =
    element publisher_info {
        publisher_info_attlist,
        ((publisher, publisher_office) | publisher | publisher_office),
        publisher_other_info?
    }
publisher_info_attlist &= empty
publisher =
    element publisher { publisher_attlist, publisher_name, address_info? }
publisher_attlist &= empty
publisher_name =
    element publisher_name { publisher_name_attlist, TextWithGaiji }
publisher_name_attlist &= attribute reading { text }?

publisher_office =
    element publisher_office {
        publisher_office_attlist, organization_name, address_info?
    }
publisher_office_attlist &= attribute publisher_code { text }?

publisher_other_info =
    element publisher_other_info {
        publisher_other_info_attlist, TextWithGaiji
    }
```

publisher_other_info_attlist &=

[a:defaultValue = "preserve"]

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

seller_info =

element seller_info {

seller_info_attlist,

((seller, seller_office) | seller | seller_office),

seller_other_info?

}

seller_info_attlist &= empty

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

seller_attlist &= empty

seller_name = element seller_name { seller_name_attlist, TextWithGaiji }

seller_name_attlist &= attribute reading { text }?

seller_office =

element seller_office {

seller_office_attlist, organization_name, address_info?

}

seller_office_attlist &= attribute seller_code { text }?

seller_other_info =

element seller_other_info { seller_other_info_attlist, TextWithGaiji }

seller_other_info_attlist &=

[a:defaultValue = "preserve"]

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

book_id_info = element book_id_info { book_id_info_attlist, book_id+ }

book_id_info_attlist &= empty

book_id = element book_id { book_id_attlist, text }

book_id_attlist &= attribute type { text }

```
classification_info =
    element classification_info {
        classification_info_attlist, classification+
    }
classification_info_attlist &= empty
classification =
    element classification { classification_attlist, TextWithGaiji }
classification_attlist &= attribute type { text }
rating = element rating { rating_attlist, empty }
rating_attlist &=
    [ a:defaultValue = "no" ] attribute adult { Yes_No }?,
    [ a:defaultValue = "no" ] attribute violence { Yes_No }?

publication_place =
    element publication_place { publication_place_attlist, text }
publication_place_attlist &= empty
publication_date_info =
    element publication_date_info {
        publication_date_info_attlist, publication_date+
    }
publication_date_info_attlist &= empty
publication_date =
    element publication_date { publication_date_attlist, text }
publication_date_attlist &=
    [ a:defaultValue = "publish" ] attribute type { "publish" | "sale" }?

net_price_info =
    element net_price_info { net_price_info_attlist, net_price+ }
net_price_info_attlist &= empty
net_price = element net_price { net_price_attlist, text }
net_price_attlist &=
    attribute country { text }?,
    attribute unit { text },
```

```
attribute other_info { text }?

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

front_cover_image =
    element front_cover_image { front_cover_image_attlist, text }
front_cover_image_attlist &= attribute type { text }

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

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

body_module =
    element body_module { body_module_attlist, flow_type_body }
body_module_attlist &= empty

flow_type_body =
    element flow_type_body {
        flow_type_body_attlist,
        flow_entry,
        special_page_link?,
        search_table?
    }
flow_type_body_attlist &= empty
```

```
flow_entry =  
    element flow_entry {  
        flow_entry_attlist, flow_default_attribute?, flow_data+  
    }  
flow_entry_attlist &= empty
```

```
flow_default_attribute =  
    element flow_default_attribute {  
        flow_default_attribute_attlist,  
        flow_default_size?,  
        flow_default_font?,  
        flow_default_background?,  
        flow_default_line_breaking_method?  
    }
```

```
flow_default_attribute_attlist &=  
    attribute baseline { BaseLine }?,  
    attribute view_type { ViewType }?
```

```
flow_default_size =  
    element flow_default_size { flow_default_size_attlist, empty }
```

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

```
flow_default_font =  
    element flow_default_font { flow_default_font_attlist, empty }
```

```
flow_default_font_attlist &=  
    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" }?
flow_default_background =
    element flow_default_background { flow_default_background_attlist, empty }
flow_default_background_attlist &=
    attribute color_space { "RGB" }?,
    attribute opacity { "100" }?,
    attribute color { text }?

flow_default_line_breaking_method =
    element flow_default_line_breaking_method {
        flow_default_line_breaking_method_attlist
    }
flow_default_line_breaking_method_attlist &=
    [ a:defaultValue = "none" ] attribute method { "none" }?

flow_data = element flow_data { flow_data_attlist, event_info? }
flow_data_attlist &=
    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 { special_page_link_attlist, special_page+ }
special_page_link_attlist &= empty
special_page = element special_page { special_page_attlist, text }
special_page_attlist &=
    [ 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 { search_table_attlist, search_table_def+ }
search_table_attlist &= empty
search_table_def =
  element search_table_def {
    search_table_def_attlist, enable_key_type, key_normalization
  }
search_table_def_attlist &=
  attribute id { text },
  [ a:defaultValue = "no" ] attribute use_default { Yes_No }?,
  [ a:defaultValue = "unicode" ] attribute sorting_rule { "unicode" }?
enable_key_type =
```

```

    element enable_key_type { enable_key_type_attlist, empty }
enable_key_type_attlist &=
    [ a:defaultValue = "no" ] attribute numerals { Yes_No }?,
    [ a:defaultValue = "no" ] attribute basic_alphabet { Yes_No }?
key_normalization =
    element key_normalization { key_normalization_attlist, empty }
key_normalization_attlist &=
    [ a:defaultValue = "yes" ] attribute capitalization{ Yes_No }?

event_info = element event_info { event_info_attlist, event+ }
event_info_attlist &=
    [ a:defaultValue = "single" ] attribute display_type { "single" }?
event = element event { event_attlist, trigger, action }
event_attlist &= empty
trigger = element trigger { trigger_attlist, Trigger_List }
trigger_attlist &= empty
action = element action { action_attlist, Action_List }
action_attlist &=
    [ a:defaultValue = "sequential" ]
    attribute action_flag { "sequential" }?

trigger_pointer =
    element trigger_pointer { trigger_pointer_attlist, pointer_region? }
trigger_pointer_attlist &=
    attribute id { text },
    [ a:defaultValue = "click" ] attribute action_flag { "click" }?
pointer_region =
    element pointer_region { pointer_region_attlist, vertex+ }
pointer_region_attlist &= empty

action_play = element action_play { action_play_attlist, empty }
action_play_attlist &=

```

```
attribute object_id { text },
[ a:defaultValue = "normal" ] attribute action { "normal" }?,
[ a:defaultValue = "no" ] attribute play_flag { "no" }?,
[ a:defaultValue = "normal" ] attribute effect { "normal" }?
action_page_jump =
    element action_page_jump { action_page_jump_attlist, empty }
action_page_jump_attlist &=
    attribute book { text }?,
    attribute book_type { text }?,
    attribute page_id { text }?,
    attribute center { text }?,
    [ a:defaultValue = "normal" ] attribute show { "normal" }?,
    [ a:defaultValue = "default" ] attribute close_effect { "default" }?,
    [ a:defaultValue = "default" ] attribute show_effect { "default" }?

parts_module =
    element parts_module { parts_module_attlist, object_table }
parts_module_attlist &= empty
object_table =
    element object_table {
        object_table_attlist,
        (dynamic_text_object_entry
        | sound_object_entry
        | search_page_object_entry
        | movie_object_entry,
        | comic_object_entry )+
    }
object_table_attlist &= empty

dynamic_text_object_entry =
    element dynamic_text_object_entry {
        dynamic_text_object_entry_attlist, permission_info?, protection?
```

```
    }  
dynamic_text_object_entry_attlist &=  
    attribute src { text },  
    attribute type { text },  
    attribute object_id { text }  
  
sound_object_entry =  
    element sound_object_entry {  
        sound_object_entry_attlist, protection?, sound_object_info  
    }  
sound_object_entry_attlist &=  
    attribute src { text },  
    attribute type { text }  
sound_object_info =  
    element sound_object_info { sound_object_info_attlist, empty }  
sound_object_info_attlist &= attribute object_id { text }  
  
search_page_object_entry =  
    element search_page_object_entry {  
        search_page_object_entry_attlist, permission_info?, protection?  
    }  
search_page_object_entry_attlist &=  
    attribute src { text },  
    attribute type { text },  
    attribute object_id { text }  
  
movie_object_entry =  
    element movie_object_entry {  
        movie_object_entry_attlist, protection?  
    }  
movie_object_entry_attlist &=  
    attribute src { text },
```

attribute type { text },
attribute object_id { text }

comic_object_entry =

element comic_object_entry {
 comic_object_entry_attlist, permission_info?, protection?
}

comic_object_entry_attlist &=

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

text_data =

element text_data {
 text_data_attlist, text_default_attribute, text_body
}

text_data_attlist &= empty

text_default_attribute =

element text_default_attribute {
 text_default_attribute_attlist,
text_default_font?,
text_default_background?,
text_default_background_music?,
line_breaking_method?
}

text_default_attribute_attlist &=

attribute baseline { BaseLine }?,
attribute valign { "middle" }?

text_default_font =

element text_default_font { text_default_font_attlist, empty }

text_default_font_attlist &=

```
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 {  
        text_default_background_attlist,  
        permission_info?  
    }
```

```
text_default_background_attlist &=
```

```
    attribute color_space { "RGB" }?,  
    attribute opacity { "100" }?,  
    attribute color { text }?,  
    (attribute src { text },  
    attribute type { text })?
```

```
text_default_background_music =
```

```
    element text_default_background_music {  
        text_default_background_music_attlist,  
        permission_info?  
    }
```

```
text_default_background_music_attlist &=
```

```
    attribute src {text},  
    attribute type {text},  
    [ a:defaultValue = "yes" ] attribute loop{ Yes_No }?
```

```
line_breaking_method =
```

```
    element line_breaking_method {  
        line_breaking_method_attlist  
    }
```

```
line_breaking_method_attlist &=
```

```
    attribute method { "none" }?
```

Block = p | scrolling_text | key_entry

Inline =

br
| hr
| font
| horizontal
| ruby
| external_char
| mask
| char_id

All_tag2 = text | Block | Inline | object

All_tag = All_tag2 | page_break

text_body = element text_body { text_body_attlist, All_tag* }

text_body_attlist &= empty

p = element p { p_attlist, All_tag2* }

p_attlist &=

attribute top_line_indent { text }?,
attribute top { text }?,
attribute bottom { text }?,
attribute align { "center" | "right" | "left" }?,
[a:defaultValue = "1"] attribute drop_cap { text }?

scrolling_text =

element scrolling_text {
scrolling_text_attlist, (text | external_char | font | horizontal | ruby | object)*
}

scrolling_text_attlist &= empty

page_break = element page_break { page_break_attlist, empty }

page_break_attlist &=

attribute turning_page_control { Turn_Page_Val }?

br = element br { br_attlist, empty }

br_attlist &= attribute clear { "left" | "right" | "all" }?

hr = element hr { hr_attlist, empty }

hr_attlist &=

attribute size { text }?,

attribute length { text }?,

attribute align { "left" | "center" | "right" }?

font = element font { font_attlist, (text | Inline | object)* }

font_attlist &=

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 }?

horizontal = element horizontal { horizontal_attlist, TextWithGaiji }

horizontal_attlist &= empty

ruby = element ruby { ruby_attlist, rbase, rtop }

ruby_attlist &= empty

rbase = element rbase { rbase_attlist, TextWithGaiji }

rbase_attlist &= empty

rtop = element rtop { rtop_attlist, TextWithGaiji }

rtop_attlist &= empty

external_char = element external_char { external_char_attlist, text }

external_char_attlist &=

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 {  
    mask_attlist,  
    (text | br | font | horizontal | ruby | external_char | object)*  
  }
```

```
mask_attlist &=  
  [ 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 { char_id_attlist, (text | Inline | object)* }  
char_id_attlist &= attribute char_id { text }
```

```
object =  
  element object { object_attlist, permission_info?, protection? }
```

```
object_attlist &=  
  attribute type { text },  
  attribute src { text },  
  attribute char_id { text }?,  
  attribute align { "top" | "middle" | "bottom" | "left" | "right" }?,  
  attribute start { "auto" | "event" }?,  
  attribute loop { "1" }?
```

```
key_entry = element key_entry { key_entry_attlist, key_item+ }
```

```
key_entry_attlist &= empty
```

```
key_item = element key_item { key_item_attlist, TextWithGaiji }
```

```
key_item_attlist &=
```

```
attribute search_word { text },
attribute table_id { text }
```

```
flip_animation =
```

```
element flip_animation {
    flip_animation_attlist,
    flip_animation_sound?,
    flip_animation_source+
}
```

```
flip_animation_attlist &=
```

```
[ a:defaultValue = "1s" ] attribute renewal_time { text }?
```

```
flip_animation_sound =
```

```
element flip_animation_sound { flip_animation_sound_attlist, empty }
```

```
flip_animation_sound_attlist &=
```

```
attribute type { text },
attribute src { text },
[ a:defaultValue = "0s" ] attribute start { "0s" }?
```

```
flip_animation_source =
```

```
element flip_animation_source { flip_animation_source_attlist, empty }
```

```
flip_animation_source_attlist &=
```

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

```
search_page =
```

```
element search_page {
    search_page_attlist,
    search_page_title?,
    key_input_region,
    key_input_region?,
    search_link_item*
```

```
}
```

search_page_attlist &= empty

search_page_title =

element search_page_title { search_page_title_attlist, TextWithGaiji }

search_page_title_attlist &=

attribute type { text }?,

attribute src { text }?

key_input_region =

element key_input_region {

key_input_region_attlist, key_input_region_prompt, enable_key_type

}

key_input_region_attlist &=

attribute table_id { text },

attribute search_type { "matches_first" | "matches_only" }?

key_input_region_prompt =

element key_input_region_prompt {

key_input_region_prompt_attlist, TextWithGaiji

}

key_input_region_prompt_attlist &= empty

search_link_item =

element search_link_item {

search_link_item_attlist, search_link_title

}

search_link_item_attlist &= attribute char_id { text }

search_link_title =

element search_link_title { search_link_title_attlist, TextWithGaiji }

search_link_title_attlist &= empty

comic_data =

element comic_data {

comic_data_attlist,

comic_body

```

}
comic_data_attlist &= empty
comic_body =
    element comic_body {
        comic_body_attlist,
        comic_cell_type_body
    }
comic_body_attlist &= empty
comic_cell_type_body =
    element comic_cell_type_body {
        comic_cell_type_body_attlist,
        cell +
    }
comic_cell_type_body_attlist &=
    attribute bg_color { text }?,
    attribute bg_color_space { "RGB" }?,
    attribute bg_opacity { "100" } ?
cell =
    element cell {
        cell_attlist,
        cell_scene*
    }
cell_attlist &=
    attribute src { text }?,
    attribute type { text }?,
    attribute char_id { text }?,
    attribute bg_color { text }?,
    attribute bg_color_space { "RGB" }?,
    attribute bg_opacity { "100" } ?
    [ a:defaultValue = "no" ] attribute fit { Yes_No }?,
    attribute effect = {
"fade-in" |

```

“dissolve” |
“slide-right” |
“slide-left” |
“slide-up” |
“slide-down” |
“overwrite-right” |
“overwrite-left” |
“overwrite-up” |
“overwrite-down” |
“pushout-right” |
“pushout-left” |
“pushout-up” |
“pushout-down” |
“box-center” |
“box-rightdown” |
“box-leftdown” |
“box-rightup” |
“box-leftup” |
“box-free”}?,

[a:defaultValue = "5"] attribute effect_time {"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9"|"10"}?,

attribute effect_pos {text}?,

[a:defaultValue = "no"] attribute scroll {

“custom” |

“up” |

“down” |

“left” |

“right” |

“no”

}?,

[a:defaultValue = "5"] attribute scroll_time {"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9"|"10"}?,

[a:defaultValue = "off"] attribute vibration { “on” | “off” }?,

[a:defaultValue = "3"] attribute vibration_time {"1"|"2"|"3"|"4"|"5"}?,

attribute sound {text}?,
attribute sound_type {text}?,
attribute url_jump {text}?

cell_scene =

element cell_scene {
 cell_scene_attlist
}

cell_scene_attlist &=

attribute position {text}?,
[a:defaultValue = "5"] attribute scroll_time {"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9"|"10"}?,
[a:defaultValue = "off"] attribute vibration { "on" | "off" }?,
[a:defaultValue = "3"] attribute vibration_time {"1"|"2"|"3"|"4"|"5"}?,
attribute sound {text}?,
attribute sound_type {text}?,
attribute url_jump {text}?

start = flip_animation | bvf | text_data | search_page | comic_data



Bibliography

The following documents have served as references in the preparation of this International Standard:

ISO/IEC 10179:1996, *Information technology – Processing languages – Document Style Semantics and Specification Language (DSSSL)* (available in English only)

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> (available in English only)

W3C Recommendation, *Extensible Stylesheet Language (XSL) Version 1.0*, 2001-10, <http://www.w3.org/TR/2001/REC-xsl-20011015/> (available in English only)

Internet Engineering Task Force(IETF), *RFC 3986-Uniform Resource Identifier (URI): Generic Syntax*, <http://www.ietf.org/rfc/rfc3986.txt> (available in English only)

IECNORM.COM: Click to view the full PDF of IEC 62448:2009

IECNORM.COM: Click to view the full PDF of IEC 62448:2009

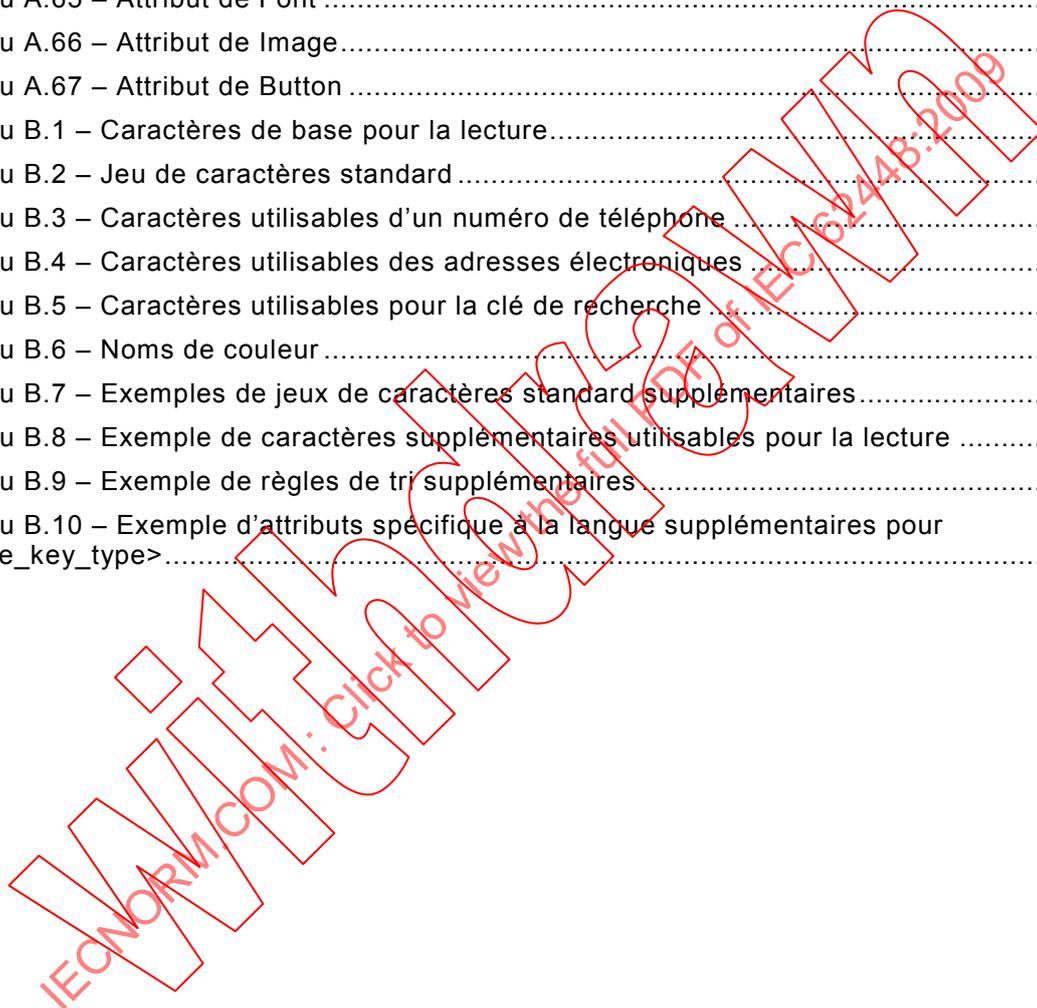
Withdrawn

SOMMAIRE

AVANT-PROPOS.....	185
INTRODUCTION.....	187
1 Domaine d'application	188
2 Références normatives.....	188
3 Termes et définitions	188
4 Position et exigences relatives au format générique	189
5 Notation	190
6 Structure logique	190
7 Sémantique	192
Annexe A (normative) Format BBeB Xylog.....	193
Annexe B (normative) Format XMDF XML.....	269
Bibliographie.....	366
Figure 1 – Modèle de création/distribution de contenu.....	189
Figure A.1 – Diagramme conceptuel de la « disposition en bloc ».....	212
Figure A.2 – Système de coordonnées.....	213
Figure A.3 – Composition de la mise en page.....	214
Figure A.4 – Composition de la zone Disposition en bloc.....	215
Figure A.5 – Attribut Bgimagemode	222
Figure A.6 – Attribut Layout	223
Figure B.1 – Structure de l'arborescence XML.....	269
Figure B.2 – Exemple de valign="middle".....	307
Figure B.3 – Exemple de lettrine.....	311
Figure B.4 – Marges de gauche et de droite d'un paragraphe	312
Figure B.5 – Écriture horizontale dans un texte vertical	315
Figure B.6 – Ruby.....	316
Figure B.7 – Exemple de rendu d'une instance d'objet de page de recherche.....	324
Figure B.8 – Instance d'objet d'illustration.....	328
Tableau A.1 – Types de données.....	216
Tableau A.2 – Attribut de BBeBXylog.....	216
Tableau A.3 – Attribut de l'élément Info	217
Tableau A.4 – Attribut de Title	217
Tableau A.5 – Attribut de l'élément Author	217
Tableau A.6 – Attribut de Publisher.....	218
Tableau A.7 – Attribut de Label.....	218
Tableau A.8 – Attribut de Cthumbnail.....	219
Tableau A.9 – Attribut de TocLabel.....	219
Tableau A.10 – Attribut de Page	220
Tableau A.11 – Attribut de Wait	224
Tableau A.12 – Attribut de TextBlock	224
Tableau A.13 – Attribut rubyalign	227

Tableau A.14 – Attribut rubyadjust	227
Tableau A.15 – Attribut rubyoverhang	227
Tableau A.16 – Attribut empdotsposition	228
Tableau A.17 – Attribut emplineposition	228
Tableau A.18 – Attribut emplinetype	228
Tableau A.19 – Attribut align	229
Tableau A.20 – Attribut blockrule	230
Tableau A.21 – Attribut de P	231
Tableau A.22 – Attribut de Plot	231
Tableau A.23 – Attribut de Fill	232
Tableau A.24 – Attribut de Space	232
Tableau A.25 – Attribut de CharButton	233
Tableau A.26 – Attribut de Gaiji	233
Tableau A.27 – Attribut de Box	234
Tableau A.28 – Attribut linetype	234
Tableau A.29 – Attribut d'EmpDots	234
Tableau A.30 – Attribut de Empline	235
Tableau A.31 – Attribut de DrawChar	236
Tableau A.32 – Attribut de ImageBlock	236
Tableau A.33 – Attribut de ButtonBlock	239
Tableau A.34 – Attribut de BaseButton	241
Tableau A.35 – Attribut de FocusinButton	241
Tableau A.36 – Attribut de JumpTo	241
Tableau A.37 – Attribut de Run	242
Tableau A.38 – Attribut runoption	242
Tableau A.39 – Attribut de PushButton	242
Tableau A.40 – Attribut de UpButton	243
Tableau A.41 – Attribut de Canvas	243
Tableau A.42 – Attribut de PutObj	244
Tableau A.43 – Attribut de Moveto	245
Tableau A.44 – Attribut de Lineto	245
Tableau A.45 – Attribut arrowtype	246
Tableau A.46 – Attribut de DrawBox	246
Tableau A.47 – Attribut de DrawEllipse	247
Tableau A.48 – Attribut de RuledLine	248
Tableau A.49 – Attribut de BookStyle	248
Tableau A.50 – Attribut de SetDefault	249
Tableau A.51 – Attribut setwaitprop	250
Tableau A.52 – Attribut de RegistFont	250
Tableau A.53 – Attribut de BookSetting	251
Tableau A.54 – Attribut de TextStyle	252
Tableau A.55 – Attribut de BlockStyle	255
Tableau A.56 – Attribut de PageStyle	257

Tableau A.57 – Attribut de Window	259
Tableau A.58 – Attribut de PopUpWin	260
Tableau A.59 – Attribut de Sound	260
Tableau A.60 – Attribut de SoundStream	261
Tableau A.61 – Attribut de ImageStream.....	261
Tableau A.62 – Attribut de Header	262
Tableau A.63 – Attribut de Footer	263
Tableau A.64 – Attribut de eSound	263
Tableau A.65 – Attribut de Font	264
Tableau A.66 – Attribut de Image.....	265
Tableau A.67 – Attribut de Button	266
Tableau B.1 – Caractères de base pour la lecture.....	271
Tableau B.2 – Jeu de caractères standard.....	271
Tableau B.3 – Caractères utilisables d’un numéro de téléphone	279
Tableau B.4 – Caractères utilisables des adresses électroniques	279
Tableau B.5 – Caractères utilisables pour la clé de recherche	296
Tableau B.6 – Noms de couleur	335
Tableau B.7 – Exemples de jeux de caractères standard supplémentaires.....	336
Tableau B.8 – Exemple de caractères supplémentaires utilisables pour la lecture	336
Tableau B.9 – Exemple de règles de tri supplémentaires	338
Tableau B.10 – Exemple d’attributs spécifique à la langue supplémentaires pour <enable_key_type>.....	338



COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

**SYSTÈMES ET ÉQUIPEMENTS MULTIMÉDIA –
PUBLICATION ET LIVRES ÉLECTRONIQUES MULTIMÉDIA –
FORMAT GÉNÉRIQUE POUR LA PUBLICATION ÉLECTRONIQUE**

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 n'a prévu aucune procédure de marquage valant indication d'approbation et n'engage pas sa responsabilité pour les équipements déclarés conformes à une de ses Publications.
- 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 62448 a été établie par le domaine technique 10: Publication et livre électronique multimédia, du comité d'études 100 de la CEI: Systèmes et appareils audio, vidéo et multimédia.

Cette seconde édition annule et remplace la première édition publiée en 2007 et constitue une révision technique.

Les principales modifications par rapport à l'édition précédente sont répertoriées ci-dessous:

- Ajout de l'Annexe B – Format X MDF XML
- Modifications connexes des Articles 2 et 6.

La présente version bilingue, publiée en 2009-06, correspond à la version anglaise.

Le texte anglais de cette norme est issu des documents 100/1386A/CDV et 100/1501/RVC.

Le rapport de vote 100/1501/RVC donne toute information sur le vote ayant abouti à l'approbation de cette norme.

La version française de cette norme n'a pas été soumise au vote.

Cette publication a été rédigée selon les Directives ISO/CEI, Partie 2.

Le comité a décidé que le contenu de cette publication ne sera pas modifié avant la date de maintenance 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.

IECNORM.COM: Click to view the full PDF of IEC 62448:2009

Withdrawn

INTRODUCTION

Le marché des livres et publications électroniques multimédia nécessite de normaliser les formats d'échange de données électroniques entre les personnes concernées, à savoir les auteurs, les préparateurs de données, les éditeurs et les lecteurs. Les formats sont classés en format de soumission, en format générique et en format de lecture. Le format de soumission doit assurer l'interaction entre les auteurs et les préparateurs de données. Le format de lecture dépend des équipements de publication électronique. Le format générique doit fournir un format d'échange aux préparateurs de données et aux éditeurs. Par conséquent, il convient qu'il soit indépendant des équipements de publication électronique.

La Commission Électrotechnique Internationale (CEI) attire l'attention sur le fait que la conformité au présent document peut impliquer l'utilisation de droits de propriété intellectuelle.

Sony Corporation a informé la CEI qu'elle possède des droits de propriété intellectuelle ou a octroyé des droits de propriété intellectuelle.

La CEI ne prend pas position concernant la preuve, la validité et le domaine d'application de ce droit de propriété intellectuelle.

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 du droit de propriété est enregistrée auprès de la CEI. Des informations peuvent être obtenues auprès de:

Sony Corporation
6-7-35 Kitashinagawa, Shinagawa-ku
JP – Tokyo 141-0001

Sharp Corporation a informé la CEI qu'elle possède des droits de propriété intellectuelle ou a octroyé des droits de propriété intellectuelle.

La CEI ne prend pas position concernant la preuve, la validité et le domaine d'application de ce droit 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 du 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.

SYSTÈMES ET ÉQUIPEMENTS MULTIMÉDIA – PUBLICATION ET LIVRES ÉLECTRONIQUES MULTIMÉDIA – FORMAT GÉNÉRIQUE POUR LA PUBLICATION ÉLECTRONIQUE

1 Domaine d'application

La présente Norme internationale spécifie un format générique de publication électronique multimédia utilisé pour l'échange des données d'un livre électronique entre les préparateurs de ces données et les éditeurs, satisfaisant un certain nombre d'exigences de ces derniers: une structure logique révisable, extensible et hétérogène.

2 Références normatives

Les documents de référence suivants sont indispensables pour l'application du présent document. Pour des 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)

ISO 639 (toutes les parties), *Codes pour la représentation des noms de langue*

ISO 646, *Traitement de l'information – Jeu ISO de caractères codés à 7 éléments pour l'échange d'information*

ISO 3166-1, *Codes pour la représentation des noms de pays et de leurs subdivisions – Partie 1: Codes de pays*

ISO 8601, *Éléments de données et formats d'échange – Échange d'information – Représentation de la date et de l'heure*

ISO/CEI 10646, *Technologies de l'information – Jeu universel de caractères codés sur plusieurs octets (JUC)*

Amendement 1 (2005)

Amendement 2 (2006)

Amendement 3 (2008)

Amendement 4 (2008)

Amendement 5 (2008)

ISO/CEI 19757-2:2003, *Technologies de l'information – Langage de définition de schéma de documents (DSDL) – Partie 2: Validation de grammaire orientée courante – RELAX NG*

Amendment 1 (2006)

3 Termes et définitions

Pour les besoins du présent document, les termes et définitions suivants s'appliquent.

3.1

livre électronique multimédia

contenu multimédia composé de textes, de graphiques, de sons et/ou de vidéos

[CEI 62229, 3.2]

3.2

préparateur de données

organisation ou personne qui met un livre électronique en forme; un éditeur est un exemple de préparateur

[CEI 62229, 3.3 modifiée]

3.3

éditeur

organisation ou personne qui publie et distribue un livre électronique

[CEI 62229, 3.4]

3.4

format générique

format du contenu d'un livre électronique multimédia, créé et édité par un préparateur de données, puis transformé par un éditeur dans un format de lecture permettant de distribuer le livre électronique

4 Position et exigences relatives au format générique

4.1 Format générique du modèle de création/distribution de contenu

Le modèle conceptuel de publication électronique multimédia [CEI 62229] définit un modèle de création/distribution de contenu représenté à la Figure 1.

Auteur <--(1)--> Préparateur des données <--(2)--> Editeur --(3)--> Lecteur

IEC 2379/06

Figure 1 – Modèle de création/distribution de contenu

Entre les étapes de traitement adjacent, le préparateur de données et l'éditeur, l'échange de contenu du livre électronique a lieu à l'aide du format générique. En fusionnant les textes, les graphiques, les sons et/ou les vidéos d'origine fournis par les auteurs, le préparateur de données crée et édite les données du livre électronique dans un format générique. Ces données sont enregistrées et transmises aux éditeurs.

Les éditeurs modifient le format générique en format de lecture approprié pour les schémas et dispositifs de distribution. Le format générique peut également être utilisé pour la distribution de livre électronique.

4.2 Exigences relatives au format générique

Le format générique peut satisfaire les exigences du préparateur de données et de l'éditeur suivantes:

a) structure logique:

le format générique doit comporter une structure logique qui peut être aisément révisée;

b) spécification de style:

il convient de présenter les éléments logiques du format générique selon une spécification de style permettant de créer un format de lecture approprié;

c) structures hétérogènes:

les livres électroniques multimédias peuvent comporter des structures hétérogènes, telles que des éléments mathématiques, chimiques et de musique, en fonction de la nature du contenu. Il convient d'importer ces structures particulières au format générique;

d) métadonnées:

il convient que le format générique prenne en charge certaines métadonnées afin de gérer le contenu du livre électronique.

5 Notation

La structure logique d'un format générique est décrite par le schéma RELAX NG de l'ISO/CEI 19757-2.

6 Structure logique

La structure logique du format générique: ebook-g, offre un format simple et extensible ebook-g-core, ainsi que les formats existants et déjà utilisés bbebylog et xmdf. Les structures de bbebylog et de xmdf sont respectivement présentées dans l'Annexe A et l'Annexe B.

default namespace=http://tc100.iec.ch/2005/ebook/generic

start = ebook-g

ebook-g = ebook-g-core

 | external "bbebylog.rnc"

 | external "xmdf.rnc"

ebook-g-core = meta-g & body-g

meta-g = external "meta-g.rnc"

body-g = element body-g { body }

body =

 title*,

 foreword*,

 introduction*,

 titled-clause+,

 annex*,

 bibliography*

title = element title {

 element main { text },

 element sub { text }?

}

foreword = element foreword { text }

introduction = element introduction { block+ }

titled-clause = element clause { id, title, clause-content }

clause-content =

 (titled-clause, titled-clause+) | untitled-clause-content

untitled-clause-content = (untitled-clause, untitled-clause+) | block+

untitled-clause = element clause { id, untitled-clause-content }

referenced-document =

 element referenced-document {

 id,

 element abbrev { text },

 element title { text },

 element field { text }*,

 element url { xsd:anyURI }

```

}
annex =
  element annex {
    id,
    title,
    clause-content
  }
bibliography = element bibliography { referenced-document+ }
block = p | ol | ul | example | note | pre | float
p = element p { inline }
float = table | figure
table = external "table.rnc"
figure = external "figure.rnc"
ol =
  element ol {
    element li { id, block+ }+
  }
ul =
  element ul {
    element li { block+ }+
  }
example = element example { p+ }
note = element note { p+ }
pre = element pre { pre-content }
pre-content =
  (text
  | element var { pre-content })*
inline =
  (text
  | element code | b | i | var { inline }
  | ref
  | strong)*
ref =
  element ref {
    attribute to { xsd:IDREF }
  }
id = attribute id { xsd:ID }?
strong = element strong { text }

```

7 Sémantique

Il convient de présenter les éléments de ebook-g-core conformément aux spécifications de style appropriées. Les spécifications de style actuelles par langages de style telles que XSL ou DSSSL, ne relèvent pas du domaine d'application de la présente norme.

Les contenus de meta-g.rnc, table.rnc et figure.rnc ne sont pas spécifiés dans une architecture de ebook-g-core.

IECNORM.COM: Click to view the full PDF of IEC 62448:2009
Withdrawn

Annexe A (normative)

Format BBeB Xylog

A.1 Généralités

La présente annexe spécifie le format bbebxilog mentionné dans l'Article 6.

Ce format est destiné à deux types d'utilisateur: le producteur du contenu du livre électronique et le développeur d'outil associé. Les personnes qui produisent du contenu peuvent utiliser cette spécification comme format générique de stockage de données reposant sur la spécification de la norme internationale. Le développeur d'outil associé peut l'utiliser en référence, afin de choisir la spécification de l'équipement vidéo et définir la règle de conversion entre cette spécification et les autres formats de données, tels que tous les types de XML ou HTML.

La spécification présentée dans la présente annexe comporte les éléments suivants:

- a) «expression du livre sur papier» concernant deux présentations sur écran et caractères modifiés, etc.
- b) «expression numérique», tels que sons, saut de page et action interactive, etc.

Les éléments nécessaires aux expressions susmentionnées et leur bon déroulement sont définis dans la présente spécification.

A.2 Description de la structure

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

namespace sch = "http://www.ascc.net/xml/schematron"

DrawChar =

(element.Plot

| element.CR

| element.Fill

| element.CharButton

| element.Yoko

| element.Tate

| element.Nekase

| element.NoBR

| element.DrawChar

| element.Italic

| element.Bold

| SimpleChar1)*

SimpleChar1 =

(element.Rubi
| element.Box
| element.EmpDots
| element.EmpLine
| element.Sub
| element.Sup
| element.Space
| SimpleChar0)*

SimpleChar0 = (text | element.Gaiji | element.AltString)*

SimpleChar2 = (element.Plot | SimpleChar0)*

BBeB XML Xylog Basic

element.BBeBXylog =

element BBeBXylog {
attlist.version,
element.BookInformation,
element.Main,
element.Style,
element.Objects
}

attlist.version &=

attribute version { text }

>> a:documentation {

' Constraint: Current "version" attribute value is "1.0" '

]

element.BookInformation =

element BookInformation { element.Info, element.TOC }

element.Info =

element Info {
attlist.version, element.BookInfo, element.DocInfo, element.Keyword*
}

element.BookInfo =

```
element BookInfo {
  element.Title,
  element.Author,
  element.BookID,
  element.Publisher,
  element.Label,
  element.Category*,
  element.Classification,
  element.FreeText
}
element.Title = element Title { attlist.reading, text }
attlist.reading &= attribute reading { text }
element.Author = element Author { attlist.reading, text }
element.BookID = element BookID { text }
element.Publisher = element Publisher { attlist.reading, text }
element.Label = element Label { attlist.reading, text }
element.Category = element Category { text }
element.Classification = element Classification { text }
element.FreeText = element FreeText { text }
element.DocInfo =
element DocInfo {
  element.CThumbnail,
  element.Language,
  element.Creator,
  element.SumPage
}
element.CThumbnail = element CThumbnail { attlist.CThumbnail }
attlist.CThumbnail &= attribute file { text }
element.Language =
element Language {
  xsd:string { length = "2" }
  >> a:documentation [
```

" Constraint: Set two-letter primary language code which is specified in ISO 639."

```
]
}
element.Creator = element Creator { text }
element.SumPage = element SumPage { text }
element.Keyword = element Keyword { text }
element.TOC = element TOC { element.TocLabel* }
element.TocLabel = element TocLabel { attlist.jump, text }
element.Main = element Main { element.Page+ }
element.Page =
  element Page {
    attlist.Page,
    element.Common.Page
  }
  >> sch:pattern [
    name = "Style Check"
    "\x{a}" ~
    " "
    sch:rule [
      context = "Page"
      "\x{a}" ~
      " "
      sch:assert [
        test = "@pagestyle = //Style/PageStyle/@stylelabel"
        "No PageStyle"
      ]
    ]
    "\x{a}" ~
    " "
  ]
  "\x{a}" ~
  " "
]
}
```

element.Common.Page =

```
(element.Common.Objects
| element.BlockSpace
| element.RuledLine
| element.Wait)*
```

attlist.id &= attribute objid { xsd:string }

attlist.refid &= attribute refobj { xsd:string }

attlist.refstream &= attribute refstream { xsd:string }

attlist.Page &=

```
attlist.id,
attribute pagestyle { text },
attlist.Common.Page
```

element.TextBlock =

```
element TextBlock {
attlist.TextBlock,
((element.P | element.CR)*
```

```
>> sch:pattern [
```

```
name = "Style Check"
```

```
"\x{a}" ~
```

```
" "
```

```
sch:rule [
```

```
context = "TextBlock"
```

```
"\x{a}" ~
```

```
" "
```

```
sch:assert [
```

```
test = "@blockstyle = //Style/BlockStyle/@stylelabel"
```

```
"No BlockStyle"
```

```
]
```

```
"\x{a}" ~
```

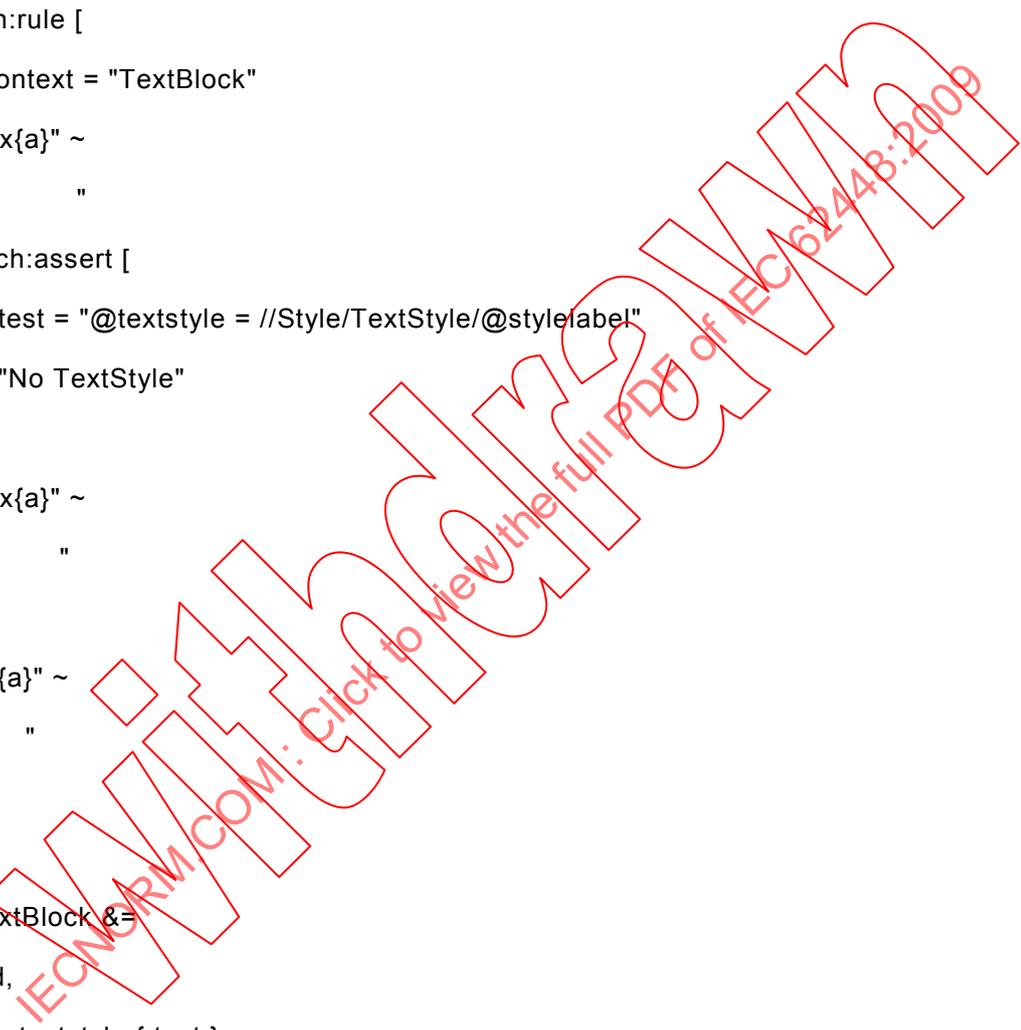
```
" "
```

```
]
```

```
"\x{a}" ~
```

```

    " "
]
>> sch:pattern [
    name = "Style Check"
    "\x{a}" ~
    " "
    sch:rule [
        context = "TextBlock"
        "\x{a}" ~
        " "
        sch:assert [
            test = "@textstyle = //Style/TextStyle/@stylelabel"
            "No TextStyle"
        ]
        "\x{a}" ~
        " "
    ]
    "\x{a}" ~
    " "
]
}
attlist.TextBlock &=
    attlist.id,
    attribute textstyle { text },
    attribute blockstyle { text },
    attlist.Common.Text,
    attlist.Common.Block
element.P = element P { attlist.P, DrawChar }
attlist.P &= attribute refesound { xsd:string }?
element.Plot = element Plot { attlist.Plot, text }
attlist.xsize &= attribute xsize { xsd:unsignedShort }
attlist.ysize &= attribute ysize { xsd:unsignedShort }
```



```

attlist.Plot &=
    attlist.xsize,
    attlist.ysize,
    attlist.refid,
    [ a:defaultValue = "bottom" ]
    attribute adjustment { "center" | "baseline" | "top" | "bottom" }?
element.CR = element CR { empty }
element.Fill =
    element Fill {
        attribute code { text },
        empty
    }
element.Space = element Space { attlist.xsize, empty }
element.CharButton = element CharButton { attlist.refid, SimpleChar1 }
element.Rubi = element Rubi { (element.Oyamoji, element.Rubimoji)+ }
element.Oyamoji = element Oyamoji { SimpleChar0 }
element.Gaiji = element Gaiji { attlist.Gaiji, text }
attlist.Gaiji &=
    attlist.refid,
    attribute fontfacename { text }?,
    attribute code { text }
element.AltString = element AltString { element.Org, element.Alt }
element.Org = element Org { text }
element.Alt = element Alt { text }
element.Rubimoji = element Rubimoji { SimpleChar0 }
element.Box = element Box { attlist.Box, SimpleChar0 }
attlist.linetype &=
    attribute linetype { "solid" | "dotted" | "dashed" | "double" }
attlist.Box &= attlist.linetype?
element.EmpDots = element EmpDots { attlist.EmpDots, SimpleChar0 }
attlist.Empdotsposition &=
    attribute empdotsposition { "before" | "after" }

```

```
attlist.EmpdotsChar &=
  attribute empdotscode { text },
  attribute empdotsfontname { text },
  attribute refempdotsfont { text }
attlist.EmxDots &= attlist.EmxDotsposition?, attlist.EmpdotsChar?
element.EmLine = element EmLine { attlist.EmLine, SimpleChar0 }
attlist.EmLineposition &=
  attribute emlineposition { "before" | "after" }
attlist.EmLinetype &=
  attribute emlinetype {
    "none" | "solid" | "dotted" | "dashed" | "double"
  }
attlist.EmLine &= attlist.EmLinetype?, attlist.EmLineposition?
element.Sub = element Sub { SimpleChar0 }
element.Sup = element Sup { SimpleChar0 }
element.Yoko = element Yoko { SimpleChar0 }
element.Tate = element Tate { SimpleChar2 }
element.Nekase = element Nekase { SimpleChar2 }
element.NoBR = element NoBR { SimpleChar1 }
element.DrawChar =
  element DrawChar {
    attribute line { xsd:unsignedShort },
    SimpleChar0
  }
element.Italic = element Italic { DrawChar }
element.Bold = element Bold { DrawChar }
element.ImageBlock =
  element ImageBlock { attlist.ImageBlock, text }
>> sch:pattern [
  name = "Style Check"
  "\x{a}" ~
  " "
```

```

sch:rule [
  context = "Page"
  "\x{a}" ~
  " "
sch:assert [
  test = "@blockstyle = //Style/BlockStyle/@stylelabel"
  "No BlockStyle"
]
"\x{a}" ~
" "
]
"\x{a}" ~
" "
]
attlist.ImageBlock &=
  attlist.id,
  attribute blockstyle { text },
  attlist.Common.Block,
  attlist.Common.Image
element.ButtonBlock =
  element ButtonBlock {
    attlist.ButtonBlock,
    element.BaseButton?,
    element.FocusinButton?,
    element.PushButton?,
    (element.UpButton?)
  }
>> sch:pattern [
  name = "Style Check"
  "\x{a}" ~
  " "
  sch:rule [
    context = "ButtonBlock"

```

```
"\x{a}" ~
"  "

sch:assert [
  test = "@blockstyle = //Style/BlockStyle/@stylelabel"
  "No BlockStyle"
]
"\x{a}" ~
"  "
]
"\x{a}" ~
"  "
]
}

attlist.ButtonBlock &=
  attlist.id,
  attribute blockstyle { text },
  attlist.Common.Block

attlist.refimage &= attribute refimage { xsd:string }
element.BaseButton = element BaseButton { attlist.refimage?, empty }
element.FocusinButton =
  element FocusinButton {
    attlist.refimage?,
    (element.JumpTo
    | element.Run
    | element.SoundStop
    | element.CloseWindow)*
  }

element.JumpTo = element JumpTo { attlist.jump, empty }
attlist.jump &=
  attribute refpage { xsd:string },
  attlist.refid
element.Run = element Run { attlist.Run, empty }
```

attlist.Run &=

[a:defaultValue = "normal"]

attribute runoption {

"normal" | "opposite" | "center" | "opposite-center"

}?,

attlist.refid

element.SoundStop = element SoundStop { empty }

element.CloseWindow = element CloseWindow { empty }

element.PushButton =

element PushButton {

attlist.refimage?,

(element.JumpTo

| element.Run

| element.SoundStop

| element.CloseWindow)*

}

element.UpButton =

element UpButton {

attlist.refimage?,

(element.JumpTo

| element.Run

| element.SoundStop

| element.CloseWindow)*

}

element.BlockSpace = element BlockSpace { attlist.BlockSpace, empty }

attlist.BlockSpace &=

attribute xspace { xsd:short }?,

attribute yspace { xsd:short }?

element.Canvas =

element Canvas {

attlist.Canvas, (element.Common.Canvas | element.Wait)*

}

element.Common.Canvas =

- element.PutObj
- | element.Moveto
- | element.Lineto
- | element.DrawBox
- | element.DrawEllipse

attlist.Canvas &=

- attribute canvaswidth { xsd:unsignedShort },
- attribute canvasheight { xsd:unsignedShort },
- attribute blockrule { "block-fixed" | "block-adjustable" }?,
- attlist.id,
- attlist.Common.Canvas

attlist.Layout &= attribute layout { "LrTb" | "TbRl" }

attlist.Common.Frame &=

- attribute framewidth { xsd:unsignedShort }?,
- attribute framecolor { text }?,
- attribute framemode { "curve" | "square" }?

attlist.Common.Canvas &=

- attribute bgcolor { text }?,
- attlist.Layout?,
- attlist.Common.Frame

element.PutObj = element PutObj { attlist.PutObj, empty }

attlist.x1.unsigned &= attribute x1 { xsd:unsignedShort }

attlist.x1.signed &= attribute x1 { xsd:short }

attlist.y1.unsigned &= attribute y1 { xsd:unsignedShort }

attlist.y1.signed &= attribute y1 { xsd:short }

attlist.PutObj &=

- attlist.x1.unsigned, attlist.y1.unsigned, attlist.refid

element.Wait =

- element Wait {
- attribute time { xsd:short },
- empty

```
}  
element.Moveto =  
    element Moveto { attlist.x1.signed, attlist.y1.signed, empty }  
element.Lineto = element Lineto { attlist.Lineto, empty }  
attlist.Common.line &=  
    attribute linewidth { xsd:unsignedShort }?,  
    attribute linecolor { text }?,  
    attlist.linetype?  
attlist.Lineto &=  
    attlist.x1.signed,  
    attlist.y1.signed,  
    attlist.Common.line,  
    attribute arrowtype { text }?  
element.DrawBox = element DrawBox { attlist.DrawBox, empty }  
attlist.DrawBox &=  
    attlist.x1.signed,  
    attlist.y1.signed,  
    attlist.Common.line,  
    attribute fillcolor { text }?  
element.DrawEllipse = element DrawEllipse { attlist.DrawEllipse, empty }  
attlist.DrawEllipse &=  
    attlist.x1.signed,  
    attlist.y1.signed,  
    attlist.Common.line,  
    attribute fillcolor { text }?  
element.RuledLine = element RuledLine { attlist.RuledLine, empty }  
attlist.RuledLine &=  
    attlist.linetype,  
    attribute linewidth { xsd:unsignedShort },  
    attribute linelength { xsd:unsignedShort },  
    attribute linecolor { text }?  
element.Style =
```

```
element Style {  
    element.BookingStyle,  
    (element.TextStyle | element.BlockStyle | element.PageStyle)*  
}
```

```
element.BookingStyle =  
    element BookingStyle {  
        attlist.Style,  
        element.SetDefault?,  
        element.RegistFont*,  
        element.BookingSetting  
    }
```

```
attlist.Style &=  
    attribute stylelabel { text },  
    attlist.id  
element.SetDefault = element SetDefault { attlist.SetDefault, empty }  
attlist.Rubyoverhang &= attribute rubyoverhang { "auto" | "none" }  
attlist.Rubyalign &= attribute rubyalign { "start" | "center" }  
attlist.Rubyadjust &= attribute rubyadjust { "line-edge" | "none" }  
attlist.Setwaitprop &= attribute setwaitprop { "replay" | "noreplay" }  
attlist.SetDefault &=  
    attlist.Rubyalign,  
    attlist.Rubyoverhang,  
    attlist.Emphdotsposition,  
    attlist.Emplineposition,  
    attlist.Emplinetype,  
    attlist.EmphdotsChar,  
    attlist.Rubyadjust,  
    attlist.Setwaitprop  
element.RegistFont = element RegistFont { attlist.RegistFont, empty }  
attlist.FontEncoding &= attribute encoding { "TTF" | "OTF" | "BF" }  
attlist.RegistFont &=  
    attribute fontfilename { text },
```

attribute file { text },

attribute fontname { text },

attlist.FontEncoding

element.BookSetting = element BookSetting { attlist.BookSetting, empty }

attlist.BookSetting &=

attribute bindingdirection { "Lr" | "Rl" },

attribute dpi { xsd:unsignedShort },

attribute screenwidth { xsd:unsignedShort },

attribute screenheight { xsd:unsignedShort },

[a:defaultValue = "24"] attribute colordepth { xsd:unsignedByte }?

element.TextStyle = element TextStyle { attlist.TextStyle, empty }

attlist.TextStyle &= attlist.Style, attlist.Common.Text

attlist.Common.Text &=

attlist.EmxDots,

attlist.EmxLine,

attribute fontsize { xsd:short }?,

attribute fontwidth { xsd:short }?,

attribute fontescapement { xsd:short }?,

attribute fontorientation { xsd:short }?,

attribute fontfacename { text }?,

attribute textcolor { text }?,

attribute textbgcolor { text }?,

attribute wordspace { xsd:short }?,

attribute letterspace { xsd:short }?,

attribute baselineskip { xsd:short }?,

attribute linespace { xsd:short }?,

attribute parindent { xsd:short }?,

attribute parskip { xsd:short }?,

attlist.Rubyalign?,

attlist.Rubyoverhang?,

attribute column { text }?,

attribute columnsep { text }?,

attribute align { "head" | "center" | "foot" }?,
attribute linecolor { text }?,
attribute charspace { xsd:short }?,
attribute fontweight { xsd:unsignedShort }?,
attlist.Rubyadjust?,
attribute textlinewidth { xsd:unsignedShort }?
element.BlockStyle = element BlockStyle { attlist.BlockStyle, empty }
attlist.BlockStyle &= attlist.Style, attlist.Common.Block
attlist.Common.Block &=
attribute blockwidth { xsd:unsignedShort }?,
attribute blockheight { xsd:unsignedShort }?,
attribute blockrule {
"horz-fixed"
| "horz-adjustable"
| "vert-fixed"
| "vert-adjustable"
| "block-fixed"
| "block-adjustable"
}?,
attribute bgcolor { text }?,
attlist.Layout?,
attlist.Common.Frame,
attribute topskip { xsd:unsignedShort }?,
attribute sidemargin { xsd:unsignedShort }?,
attribute footskip { xsd:unsignedShort }?,
attribute refbgimage { text }?,
attlist.Bgimagemode?
element.PageStyle = element PageStyle { attlist.PageStyle, empty }
attlist.PageStyle &=
attlist.Style,
attlist.Common.Page,
attribute setemptyview { "show" | "empty" }?,

attlist.Setwaitprop?

attlist.Bgimagemode &=

attribute bgimagemode { "fix" | "tile" | "centering" }

attlist.Common.Page &=

(attribute refbgimage { text },

attlist.Bgimagemode)?,

attribute evenfooterid { xsd:string }?,

attribute evenheaderid { xsd:string }?,

attribute oddfooterid { xsd:string }?,

attribute oddheaderid { xsd:string }?,

attribute pageposition { "upper" | "lower" | "any" }?,

attribute topmargin { xsd:short }?,

attribute headheight { xsd:short }?,

attribute headsep { xsd:short }?,

attribute oddsidemargin { xsd:unsignedShort }?,

attribute evensidemargin { xsd:unsignedShort }?,

attribute textheight { xsd:unsignedShort }?,

attribute textwidth { xsd:unsignedShort }?,

attribute footheight { xsd:short }?,

attlist.Layout?,

attribute footspace { xsd:short }?

element.Common.Objects &=

element.ImageBlock

| element.TextBlock

| element.ButtonBlock

| element.Canvas

element.Objects =

element Objects {

(element.Common.Objects

| element.Window

| element.PopUpWin

| element.Sound

```
| element.SoundStream
| element.ImageStream
| element.Header
| element.Footer
| element.eSound
| element.Font

| element.Image
| element.Button)*
}
element.Window = element Window { attlist.Window, element.Common.Page }
attlist.Window &=
  attribute windowwidth { xsd:unsignedShort }?,
  attribute windowheight { xsd:unsignedShort }?,
  attlist.Layout?,
  attlist.id
element.PopUpWin =
  element PopUpWin {
    attlist.PopUpWin, (element.TextBlock | element.ImageBlock)
  }
attlist.PopUpWin &= attlist.id
element.Sound = element Sound { attlist.Sound, empty }
attlist.Sound &=
  [ a:defaultValue = " 1" ] attribute times { xsd:unsignedShort }?,
  [ a:defaultValue = "sync" ] attribute playmode { "sync" | "async" }?,
  attlist.refstream,
  attlist.id
element.SoundStream = element SoundStream { attlist.SoundStream, empty }
attlist.SoundStream &=
  attribute encoding { "PCM" | "MP3" },
  attribute file { text },
  attlist.id
```

element.ImageStream = element ImageStream { attlist.ImageStream, text }

attlist.ImageStream &=

attribute encoding { "JPEG" | "GIF" | "BMP" | "PNG" },

attribute file { text },

attlist.id

element.Header =

element Header {

attlist.id, attlist.Common.Canvas, (element.Common.Canvas)*

}

element.Footer =

element Footer {

attlist.id, attlist.Common.Canvas, (element.Common.Canvas)*

}

element.eSound = element eSound { attlist.eSound, empty }

attlist.eSound &= attlist.id, attlist.refstream

element.Font = element Font { attlist.Font, empty }

attlist.Font &=

attribute file { text },

attribute fontfilename { text },

attribute fontname { text },

attlist.id,

attlist.FontEncoding

element.Image = element Image { attlist.Image, text }

attlist.Image &= attlist.id, attlist.Common.Image

attlist.Common.Image &=

attribute x0 { xsd:unsignedShort },

attribute y0 { xsd:unsignedShort },

attlist.x1.unsigned,

attlist.y1.unsigned,

attlist.xsize,

attlist.ysize,

attlist.refstream

```

element.Button =
  element Button {
    attlist.Button,
    element.BaseButton?,
    element.FocusinButton?,
    element.PushButton?,
    element.UpButton?
  }
attlist.Button &= attlist.id
start = element.BBeBXylog
    
```

A.3 Disposition en bloc

A.3.1 Disposition

Le format « BBeB Book » s'appuie sur le concept de « disposition en bloc ». Dans la « disposition en bloc », le contenu à visualiser est affiché dans une zone appelée « bloc ». La disposition en elle-même est composée de plusieurs « blocs », chacun d'eux étant placé dans un emplacement relatif. Le format de fichier « BBeB XML Xylog » définit les règles de mise en œuvre reposant sur ce concept de « disposition en bloc », tel que présenté à la Figure A.1.

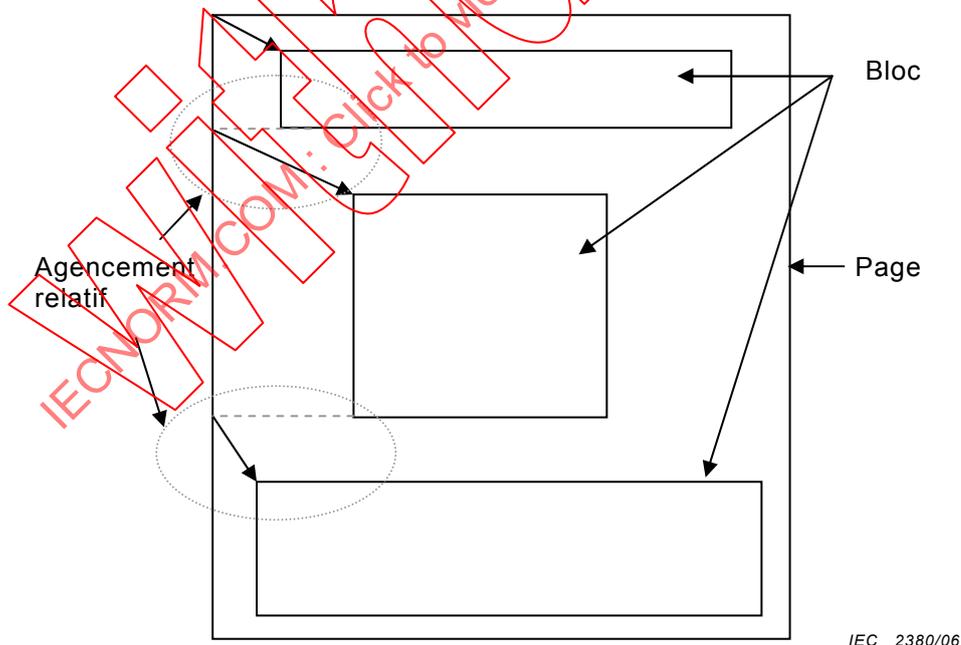


Figure A.1 – Diagramme conceptuel de la « disposition en bloc »

A.3.2 Bloc

La présente spécification définit les éléments **TextBlock**, **ImageBlock**, **ButtonBlock** et **Canvas**. Ces éléments sont placés sur une « Page » sous forme de « bloc ». Ils sont

présentés comme des sous-éléments de l'élément **Page**, l'élément **BlockSpace** indiquant l'agencement relatif.

Il est possible de spécifier l'image d'arrière-plan, la couleur d'arrière-plan et le cadre dans le « bloc ».

A.3.3 Système de coordonnées

La présente norme utilise un système de coordonnées à deux dimensions (X,Y) qui dépend de l'orientation verticale ou horizontale (voir Figure A.2).

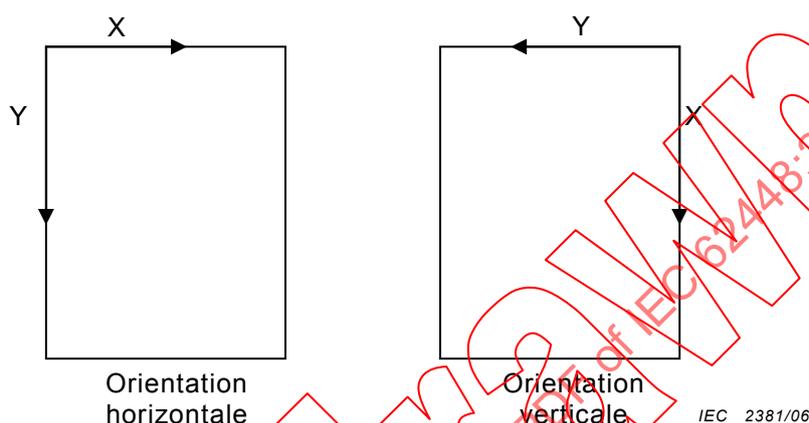


Figure A.2 – Système de coordonnées

A.4 Terminologie

A.4.1 Mise en page

Les termes suivants sont définis pour la zone de mise en page (voir Figure A.3):

Taille d'écran:

dimension de la page à afficher sur un dispositif d'affichage particulier. Elle est spécifiée par la valeur des attributs **screenwidth** et **screenheight** de l'élément **BookSetting** (voir A.5.71)

marge supérieure:

espace entre le bord supérieur de l'écran et la zone d'en-tête

zone En-tête:

zone dans laquelle est/sont placé(s) le/les en-tête(s)

hauteur d'en-tête:

hauteur de la zone En-tête

séparation d'en-tête:

espace entre la zone En-tête et la zone Texte principal

Zone Texte principal:

zone contenant le texte principal

Hauteur de texte:

hauteur de la zone Texte principal

largeur de texte:

largeur de la zone En-tête, de la zone Texte principal et de la zone Bas de page

espace de bas de page:

distance entre le bas de la zone Texte principal et le bord inférieur de la zone Bas de page

hauteur de bas de page:

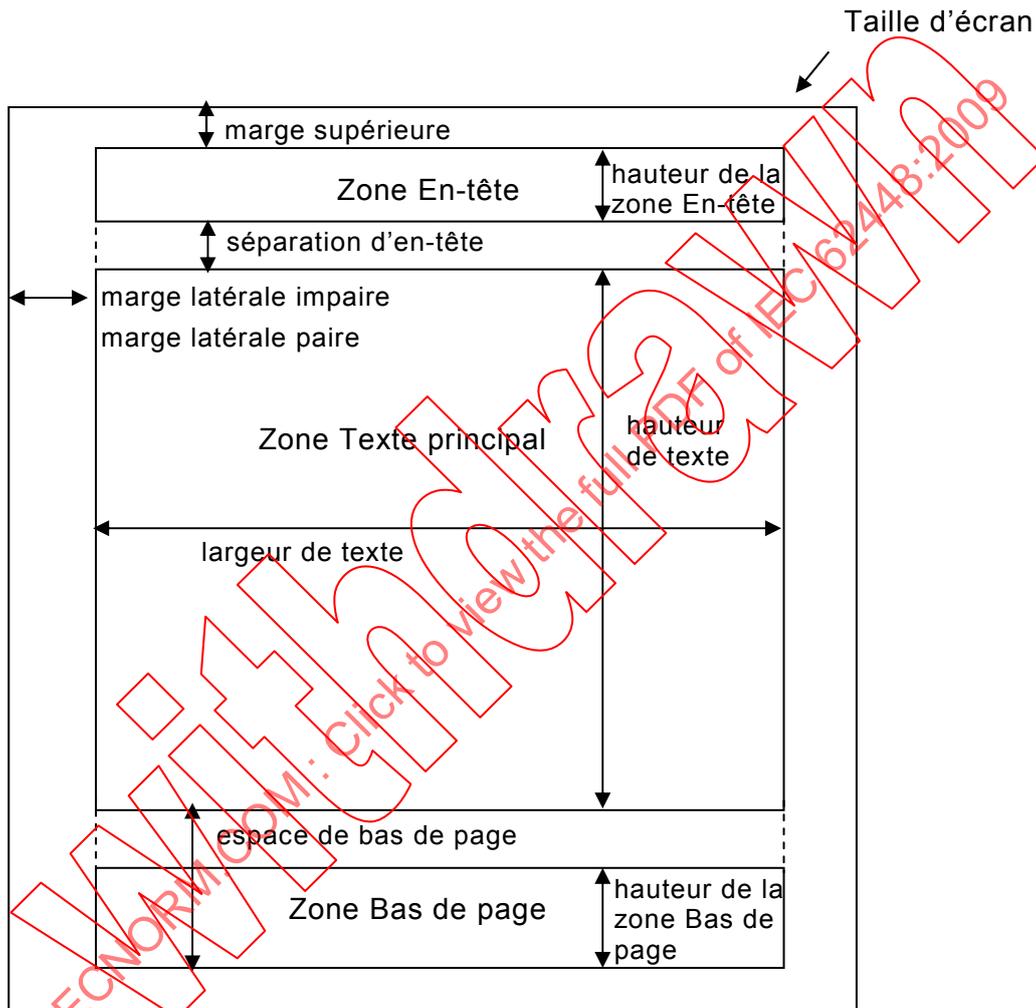
hauteur de la zone Bas de page

marge latérale impaire:

marge entre le bord gauche de l'écran et le bord gauche de la zone Texte principal de la page impaire

marge latérale paire:

marge entre le bord gauche de l'écran et le bord gauche de la zone Texte principal de la page paire



IEC 2382/06

Figure A.3 – Composition de la mise en page

A.4.2 Disposition en bloc

Les termes suivants sont définis pour la zone Disposition en bloc (voir Figure A.4):

zone de restitution:

zone définie dans un bloc et contenant le texte ou l'image

saut supérieur:

marge entre le bord supérieur du bloc et le bord supérieur de la zone de restitution

marge latérale:

marge entre le bord latéral d'un bloc et le bord latéral de la zone de restitution pour les deux côtés

saut de bas de page:

marge entre le bord inférieur du bloc et le bord inférieur de la zone de restitution

hauteur de bloc:

hauteur de la zone « Bloc »

largeur de bloc:

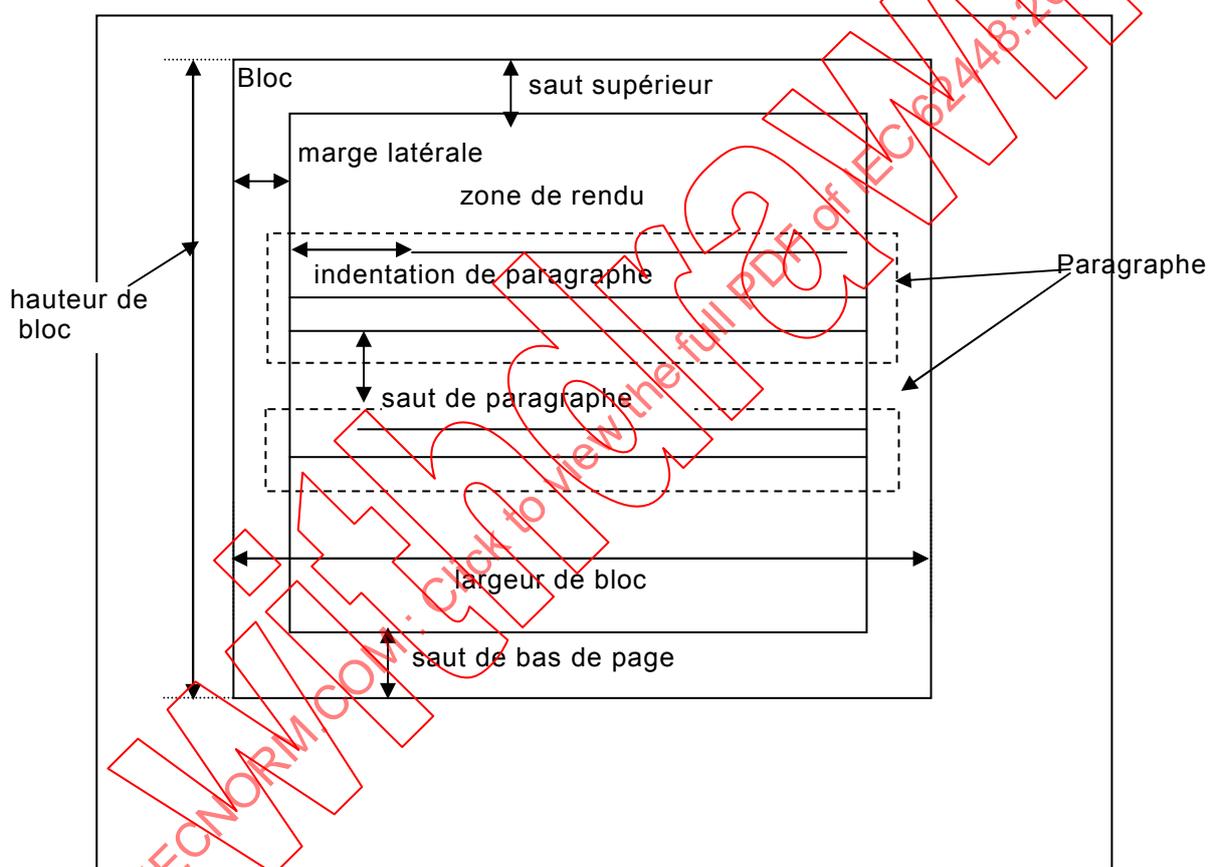
largeur de la zone « Bloc »

indentation de paragraphe:

retrait du paragraphe, mesuré à partir du bord latéral de la zone de rendu

saut de paragraphe:

distance entre la dernière ligne d'un paragraphe et la première ligne du paragraphe suivant



IEC 2383/06

Figure A.4 – Composition de la zone Disposition en bloc

A.4.3 Types de données

Les définitions de type suivantes sont définies comme la valeur XML "Attribute" dans la présente norme.

Tableau A.1 – Types de données

Définition de type	Signification	Exemple
Chaîne	Indique une chaîne de caractères normale	« après », « solide »
Chaîne de tri	Permet uniquement d'utiliser la chaîne de caractères définie dans le jeu de caractères pour la spécification de tri.	"Kennedy John F."
Chemin d'accès au fichier	Indique le chemin permettant d'accéder au fichier	"D:\sample.jpg", "..\data\white.mp3"
Chaîne décimale	Indique un nombre décimal comme chaîne de caractères	"10", "20"
Chaîne hexadécimale	Indique un nombre hexadécimal comme chaîne de caractères. "0x" est placé au début du nombre pour indiquer qu'il est hexadécimal	"0x12", "0x3333"
Chaîne de style	Indique une chaîne de caractères utilisée dans les étiquettes de style	"Main Block", "Cover Page", "Section Title Text"
Chaîne d'ID objet	Indique une chaîne de caractères utilisée dans les ID objet	"33", "Object12"
Chaîne COLORREF	Indique une chaîne de caractères hexadécimaux de 32 bits signifiant une couleur. Le format de description est 0xaaarrgbb. (aa: canal alpha [spécifié sous la forme "00" ou "ff" dans ces spécifications], "rr": valeur du rouge, "gg": valeur du vert, "bb": valeur du bleu)	"0x00ff0000" (rouge)

A.5 Éléments et attributs

A.5.1 BBeBXylog

BBeBXylog est un élément indiquant la zone de l'ensemble du contenu conformément au « format de fichier BBeB XML Xylog ».

Tableau A.2 – Attribut de BBeBXylog

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
version (obligatoire)		Chaîne	Définit comme "1.0".

L'attribut **version** indique le numéro de version correspondant du format de fichier BBeB XML Xylog. La valeur doit être "1.0" dans la présente norme.

A.5.2 BookInformation

L'élément **BookInformation** précise les informations bibliographiques et le contenu de la table des matières.

A.5.3 Info

L'élément **Info** précise les informations bibliographiques et les informations supplémentaires du contenu.

Tableau A.3 – Attribut de l'élément Info

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
version (obligatoire)		Chaîne	Définit comme "1.0"

L'attribut **version** précise la version du format des informations bibliographiques. La valeur doit être "1.0", conformément à la présente norme.

A.5.4 BookInfo

L'élément **BookInfo** précise les informations bibliographiques du contenu.

A.5.5 Title

L'élément **Title** précise le titre du contenu.

Tableau A.4 – Attribut de Title

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
reading (obligatoire)		Chaîne de tri	

L'attribut **reading** précise la chaîne permettant de trier le contenu par titre. "Japanese syllabary character", etc. peut être utilisé en japonais. "Alphabet", etc. peut être utilisé en anglais. Les caractéristiques des jeux de caractères utilisés dans cet attribut peuvent être spécifiées dans les parties relatives à l'interchangeabilité.

A.5.6 Author

L'élément **Author** précise le nom de l'auteur du contenu. Si l'utilisateur souhaite indiquer plusieurs noms d'auteur, il doit les indiquer en série dans un élément.

Tableau A.5 – Attribut de l'élément Author

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
reading (obligatoire)		Chaîne de tri	

L'attribut **reading** précise la chaîne permettant de trier le contenu par auteur. "Japanese syllabary character", etc. peut être utilisé en japonais. "Alphabet", etc. peut être utilisé en anglais. Les caractéristiques des jeux de caractères utilisés dans cet attribut peuvent être spécifiées dans les parties relatives à l'interchangeabilité.

A.5.7 BookID

L'élément **BookID** précise les informations d'ID du contenu. Cet ID permet de différencier le contenu personnel du contenu commercial. L'élément BookID du contenu personnel doit être spécifié par le caractère "FB" suivi de 14 chiffres au maximum (soit 16 chiffres en tout). L'élément BookID du contenu commercial peut être spécifié dans les parties relatives à l'interchangeabilité.

A.5.8 Publisher

L'élément **Publisher** précise le nom de l'éditeur du contenu.

Tableau A.6 – Attribut de Publisher

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
reading (obligatoire)		Chaîne de tri	

L'attribut **reading** précise la chaîne permettant de trier le contenu par éditeur. "Japanese syllabary character", etc. peut être utilisé en japonais. "Alphabet", etc. peut être utilisé en anglais. Les caractéristiques des jeux de caractères utilisés dans cet attribut peuvent être spécifiées dans les parties relatives à l'interchangeabilité.

A.5.9 Label

L'élément **Label** précise le nom de l'étiquette.

Tableau A.7 – Attribut de Label

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
reading (obligatoire)		Chaîne de tri	

L'attribut **reading** précise la chaîne permettant de trier le contenu par étiquette. "Japanese syllabary character", etc. peut être utilisé en japonais. "Alphabet", etc. peut être utilisé en anglais. Les caractéristiques des jeux de caractères utilisés dans cet attribut peuvent être spécifiées dans les parties relatives à l'interchangeabilité.

A.5.10 Category

L'élément **Category** précise la nature du contenu. Il ne doit pas y avoir plus de deux éléments de ce type.

A.5.11 Classification

L'élément **Classification** précise les informations relatives au type de données incluses dans le contenu (son, image en couleur, par exemple).

A.5.12 FreeText

L'élément **FreeText** précise une description libre du contenu (récapitulatif du contenu, par exemple).

A.5.13 DocInfo

L'élément **DocInfo** précise les informations de vignette, de langue, etc.

A.5.14 Cthumbnail

L'élément **Cthumbnail** précise le nom de fichier de la vignette relative au contenu.

Tableau A.8 – Attribut de Cthumbnail

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
file (obligatoire)		Chemin d'accès au fichier	Indique le chemin d'accès absolu ou le chemin d'accès relatif accessible à utiliser en tant que vignette

L'attribut **file** indique le nom de fichier et le chemin d'accès dans lequel se trouve le fichier de la vignette.

A.5.15 Language

L'élément **Language** indique la langue principale du contenu. Il convient d'utiliser les codes de langue de l'ISO 639. Par exemple, pour le japonais: "ja".

A.5.16 Creator

L'élément **Creator** précise le nom de l'auteur ou du studio ayant créé le contenu.

A.5.17 SumPage

L'élément **SumPage** indique le nombre de "View" (pages visibles pour les conditions de l'élément **BookSetting**) du contenu.

A.5.18 Keyword

L'élément **Keyword** indique les mots clés nécessaires à la recherche du contenu.

A.5.19 TOC

L'élément **TOC** précise la table des matières.

A.5.20 TocLabel

L'élément **TocLabel** indique la chaîne de caractères présentée comme la liste de la table des matières.

Tableau A.9 – Attribut de TocLabel

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
refobj (obligatoire)		Chaîne d'ID objet	Indique le « objid » des éléments TextBlock , ImageBlock , ButtonBlock et Canvas
refpage (obligatoire)		Chaîne d'ID objet	Indique le « objid » de l'élément Page

A.5.20.1 refobj

L'attribut **refobj** indique le « objid » de l'élément spécifié comme cible « Jump ». Il convient de placer l'élément spécifié comme « objid » dans l'élément **Page** spécifié comme « refpage ».

A.5.20.2 repage

L'attribut **repage** indique le « objid » de l'élément Page contenant l'élément spécifié comme cible « Jump ».

A.5.21 Main

L'élément **Main** indique le « Contenu principal ».

A.5.22 Page

L'élément **Page** spécifie les éléments composés de "Page" et des informations relatives à sa présentation. Il est recommandé que l'attribut de l'élément **Page** décrive uniquement les informations différentes de celles de l'élément **PageStyle** spécifiées par "pagestyle".

Tableau A.10 – Attribut de Page

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
bgimagemode		Chaîne	Indique "fix", "tile" ou "centering"
evenfooterid		Chaîne d'ID objet	Indique le « objid » de l'élément Footer
evenheaderid		Chaîne d'ID objet	Indique le « objid » de l'élément Header
evensidemargin	(Entier non signé)	Chaîne décimale	[dot]
footheight	(Entier signé)	Chaîne décimale	[dot]
footspace	(Entier signé)	Chaîne décimale	[dot]
headheight	(Entier signé)	Chaîne décimale	[dot]
headsep	(Entier signé)	Chaîne décimale	[dot]
layout		Chaîne	Indique "LrTb" ou "TbRI"
Objid (obligatoire)		Chaîne	Indique la seule chaîne de caractères dans le fichier
oddfooterid		Chaîne d'ID objet	Indique le « objid » de l'élément Footer
oddheaderid		Chaîne d'ID objet	Indique le « objid » de l'élément Header
oddsidemargin	(Entier non signé)	Chaîne décimale	[dot]
pageposition		Chaîne	Indique "upper", "lower" ou "any"
Pagestyle (obligatoire)		Chaîne de style	Il est nécessaire d'associer la chaîne de caractères de style spécifiée à "stylelabel" de l'élément PageStyle en référence
refbimage		Chaîne d'ID objet	Indique le « objid » de l'élément Image En outre, il est nécessaire de préciser en même temps l'attribut bgimagemode
textheight	(Entier non signé)	Chaîne décimale	[dot]
textwidth	(Entier non signé)	Chaîne décimale	[dot]
topmargin	(Entier signé)	Chaîne décimale	[dot]

A.5.22.1 pagestyle

L'attribut **pagestyle** spécifie "stylelabel" de l'élément **PageStyle** avec les informations nécessaires pour la présentation de "Page".

A.5.22.2 refbimage

L'attribut **refbimage** spécifie le "objid" de l'élément **Image** à afficher en arrière-plan de l'élément **Page**. Si "objid" n'est pas spécifié, l'image ne s'affiche pas.

A.5.22.3 pageposition

L'attribut **pageposition** indique si la position de départ est la page de gauche ou de droite de l'affichage à deux écrans.

Upper, lower et any sont respectivement indiqués si l'affichage a lieu en haut, en bas ou après la page précédente.

L'attribut **bindingdirection** "Rl" (contenu la plupart du temps à flux vertical) de l'élément **BookSetting**:

"upper" indique que la page de début est celle de droite.

"lower" indique que la page de début est celle de gauche.

"any" indique un affichage continu à partir de la page précédente.

L'attribut **bindingdirection** "Lr" (contenu la plupart du temps à flux horizontal) de l'élément **BookSetting**:

"upper" indique que la page de début est celle de gauche.

"lower" indique que la page de début est celle de droite.

"any" indique un affichage continu à partir de la page précédente.

A.5.22.4 oddheaderid

L'attribut **oddheaderid** indique le "objid" de l'élément **Header** affiché dans la "Page" impaire (page supérieure) de l'affichage à 2 écrans.

A.5.22.5 evenheaderid

L'attribut **evenheaderid** indique le "objid" de l'élément **Header** affiché dans la "Page" paire (pages inférieures) de l'affichage à 2 écrans.

A.5.22.6 oddfooterid

L'attribut **oddfooterid** indique le "objid" de l'élément **Footer** affiché dans la "Page" impaire (page supérieure) de l'affichage à 2 écrans.

A.5.22.7 evenfooterid

L'attribut **evenfooterid** indique le "objid" de l'élément **Footer** affiché dans la "Page" paire (pages inférieures) de l'affichage à 2 écrans.

A.5.22.8 bgimagemode

L'attribut **bgimagemode** précise la manière dont il convient de placer l'"Image" spécifiée dans l'attribut **refbimage**.

La valeur est "fix": La position de l'image est à l'origine de la « zone de texte principal ».

La valeur est "tile": Les images occupent la « zone de texte principal » à partir de l'origine.

La valeur est "center": La position de l'image est le centre de la « zone de texte principal ».

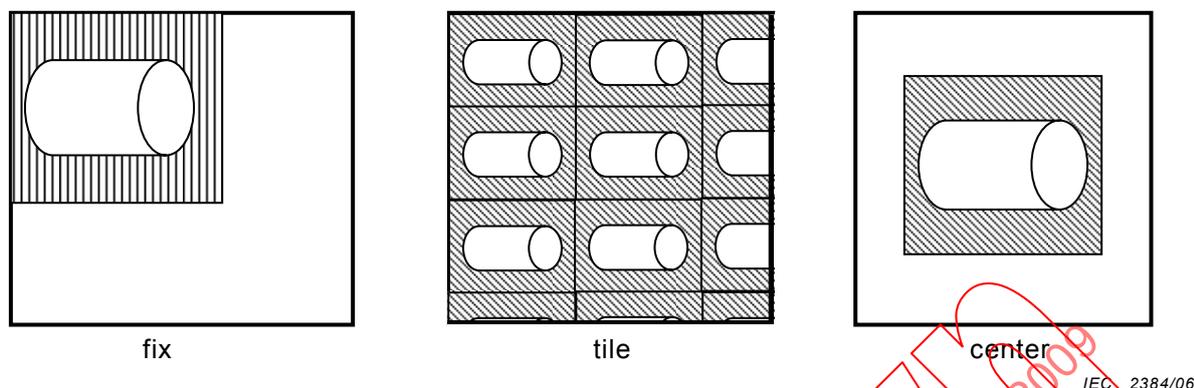


Figure A.5 – Attribut Bgimagemode

A.5.22.9 evensidemargin

L'attribut **evensidemargin** définit la marge de gauche de la « Composition de la mise en page » de l'affichage à 2 écrans: la page de gauche de la présentation horizontale et la page de droite de la présentation verticale.

A.5.22.10 footheight

L'attribut **footheight** définit la « hauteur de bas de page » de la « Composition de la mise en page ».

A.5.22.11 footspace

L'attribut **footspace** définit l'« espace de bas de page » de la « Composition de la mise en page ».

A.5.22.12 headheight

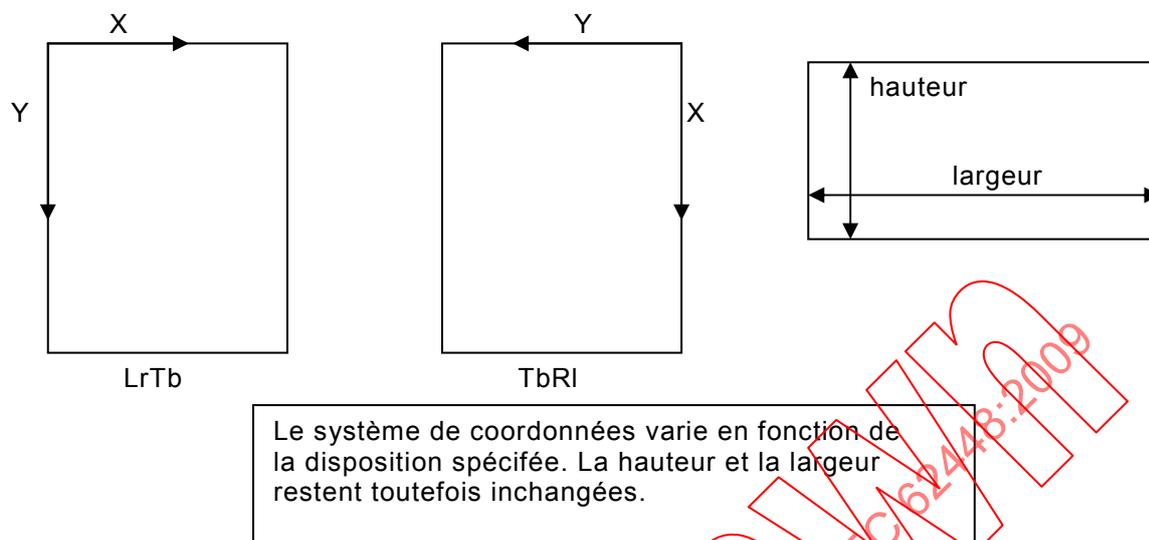
L'attribut **headheight** définit la « hauteur d'en-tête » de la « Composition de la mise en page ».

A.5.22.13 headsep

L'attribut **headsep** définit la « séparation d'en-tête » de la « Composition de la mise en page ».

A.5.22.14 layout

L'attribut **layout** spécifie le système de coordonnées de l'élément **Page**. Si "LrTb" est indiqué, l'origine se trouve en haut à gauche, l'abscisse et l'ordonnée étant respectivement orientées vers la droite et vers le bas. Si "TbRI" est indiqué, l'origine se trouve en haut à droite, l'abscisse et l'ordonnée étant respectivement orientées vers le bas et vers la gauche.



IEC 2385/06

Figure A.6 – Attribut Layout

A.5.22.15 oddsidemargin

L'attribut **oddsidemargin** définit la marge de gauche de la « Composition de la mise en page » de l'affichage à 2 écrans: la page de droite de la présentation horizontale et la page de gauche de la présentation verticale.

Il convient d'utiliser la valeur de l'attribut **oddsidemargin**. Sur l'affichage à 1 écran, la valeur de "Oddsidemargin" issue de "Evensidemargin" est utilisée.

A.5.22.16 textheight

L'attribut **textheight** définit la « hauteur de texte » de la « Composition de la mise en page ».

A.5.22.17 textwidth

L'attribut **textwidth** définit la « largeur de texte » de la « Composition de la mise en page ».

A.5.22.18 topmargin

L'attribut **topmargin** définit la « marge supérieure » de la « Composition de la mise en page ».

A.5.22.19 objid

L'attribut **objid** spécifie une chaîne de caractères unique dans le contenu de chaque élément.

A.5.23 Wait

L'élément **Wait** indique la fonction du temporisateur.

Tableau A.11 – Attribut de Wait

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
time (obligatoire)	(Entier signé)	Chaîne décimale	

L'attribut **time** spécifie le temporisateur. Si la valeur "0" est attribuée à « time », le temporisateur attend, même en cas d'opération d'entrée. Si une valeur positive est attribuée à « time », aucune action n'est entreprise pendant le temps déterminé (× 100 ms). Si une valeur négative est attribuée à « time », le temporisateur attend l'entrée pendant la valeur absolue du temps déterminé (× 100 ms).

A.5.24 TextBlock

L'élément **TextBlock** spécifie les informations de configuration de "Block" et "Text" afin d'afficher le Bloc à l'écran. L'attribut de l'élément **TextBlock** est recommandé pour décrire uniquement les informations différentes de celles de l'élément **TextStyle** et **BlockStyle** spécifiées par "textstyle" et "blockstyle".

Tableau A.12 – Attribut de TextBlock

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
align		Chaîne	Indique "head", "center" ou "foot"
baselineskip	(Entier signé)	Chaîne décimale	[pt]*10
bgcolor		Chaîne COLORREF	
bgimagemode		Chaîne	Indique "fix", "tile" ou "centering". En outre, il est nécessaire de préciser en même temps l'attribut refbgimage
blockheight	(Entier non signé)	Chaîne décimale	[dot]
blockrule		Chaîne	Indique "horz-fixed", "horz-adjustable", "vert-fixed" ou "vert-adjustable" comme attribut blockrule de l'élément TextBlock
blockstyle (obligatoire)		Chaîne de style	Il est nécessaire d'associer la chaîne de caractères de style spécifiée à "stylelabel" de l'élément BlockStyle en référence
blockwidth	(Entier non signé)	Chaîne décimale	[dot]
charspace	(Entier signé)	Chaîne décimale	[pt]*10
column		Chaîne décimale	Indique un entier compris entre 1 et 9
columnsep	(Entier signé)	Chaîne décimale	[pt]*10
empdotscode		Chaîne hexadécimale	Indique le code de caractères utilisé comme point d'emphase. En outre, il est nécessaire de préciser en même temps les attributs empdotsfontname et refempdotsfont
empdotsposition		Chaîne	Indique "before" ou "after"
emplinetype		Chaîne	Indique "none", "solid", "dotted", "double" ou "dashed"
emplineposition		Chaîne	Indique "before" ou "after"
fontescapement		Chaîne décimale	Indique 0 ou 2 700
fontfacename		Chaîne	Indique la police
fontorientation		Chaîne décimale	Indique 0 ou 2 700
fontsize	(Entier signé)	Chaîne décimale	[pt]*10

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
fontweight		Chaîne décimale	Indique une valeur comprise entre 1 et 1 000
fontwidth	(Entier signé)	Chaîne décimale	[pt]*10 Indique "-10" s'il ne modifie pas la forme de la police
footskip	(Entier non signé)	Chaîne décimale	[dot]
framecolor		Chaîne COLORREF	
framemode		Chaîne	Indique "curve" ou "square"
framewidth	(Entier non signé)	Chaîne décimale	[dot]
layout		Chaîne	Indique "LrTb" ou "TbRl"
letterspace	(Entier signé)	Chaîne décimale	[pt]*10
linecolor		Chaîne COLORREF	
linespace	(Entier signé)	Chaîne décimale	[pt]*10
textlinewidth	(Entier non signé)	Chaîne décimale	[pt]*10
objid (obligatoire)		Chaîne	Indique la seule chaîne de caractères dans le fichier
parindent	(Entier signé)	Chaîne décimale	[pt]*10
parskip	(Entier signé)	Chaîne décimale	[pt]*10
refbgimage		Chaîne d'ID objet	Indique le « objid » de l'élément Image En outre, il est nécessaire de préciser en même temps l'attribut refimagemode
rubyalign		Chaîne	Indique "start" ou "center"
rubyadjust		Chaîne	Indique la chaîne de caractères "line-edge" ou "none"
rubyoverhang		Chaîne	Indique la chaîne de caractères "auto" ou "none"
sidemargin	(Entier non signé)	Chaîne décimale	[dot]
textbgcolor		Chaîne COLORREF	
textcolor		Chaîne COLORREF	
textstyle (obligatoire)		Chaîne de style	Il est nécessaire d'associer la chaîne de caractères de style spécifiée à "stylelabel" de l'élément TextStyle en référence
toclabel			Indique la chaîne de caractères utilisée dans la liste TOC
topskip	(Entier non signé)	Chaîne décimale	[dot]
wordspace	(Entier signé)	Chaîne décimale	[pt]*10
empdotsfontname		Chaîne	Indique le nom de la police utilisée pour les points d'emphase. En outre, il est nécessaire de préciser en même temps les attributs empdotscode et refempdotsfont
refempdotsfont		Chaîne d'ID objet	Indique le « objid » de l'élément Font En outre, il est nécessaire de préciser en même temps les attributs empdotscode et empdotsname

A.5.24.1 fontsize

L'attribut **fontsize** spécifie la taille de la police. Indiqué en multiple de 10 pt.

A.5.24.2 fontwidth

L'attribut **fontwidth** spécifie la largeur de la police. En principe, la largeur optimale est fournie en fonction de la taille de la police, mais elle est utilisée lorsque la largeur est modifiée volontairement. Indiqué en multiple de 10 pt.

A.5.24.3 fontescapement

L'attribut **fontescapement** spécifie le sens de chargement du caractère. Les valeurs 0 et 2 700 sont respectivement spécifiées pour la police horizontale et la police verticale. Les autres spécifications ne sont pas acceptées.

A.5.24.4 fontorientation

L'attribut **fontorientation** spécifie le sens de rotation du caractère. Les valeurs 0 et 2 700 sont respectivement spécifiées pour la police horizontale et la police verticale. Les autres spécifications ne sont pas acceptées.

A.5.24.5 fontfacename

L'attribut **fontfacename** spécifie le nom de la police à utiliser. La police standard est utilisée si rien n'est spécifié.

A.5.24.6 textcolor

L'attribut **textcolor** spécifie la couleur du texte à utiliser. La couleur noire (0x0000000) est utilisée si rien n'est spécifié.

A.5.24.7 textbgcolor

L'attribut **textbgcolor** spécifie la couleur d'arrière-plan du texte. Un arrière-plan transparent (0xff000000) est en général utilisé si rien n'est spécifié.

A.5.24.8 wordspace

L'attribut **wordspace** spécifie la largeur des espaces entre les mots dans les langues occidentales. Indiqué en multiple de 10 pt.

A.5.24.9 letterspace

L'attribut **letterspace** spécifie la taille des espaces entre les lettres dans les langues occidentales. Indiqué en multiple de 10 pt.

A.5.24.10 charspace

L'attribut **charspace** spécifie la taille des espaces entre les caractères japonais. Indiqué en multiple de 10 pt.

A.5.24.11 baselineskip

L'attribut **baselineskip** spécifie l'espace entre les lignes. Indiqué en multiple de 10 pt.

A.5.24.12 linespace

L'attribut **linespace** spécifie l'espace garanti minimal dans le sens d'évolution des lignes. Indiqué en multiple de 10 pt.

A.5.24.13 parindent

L'attribut **parindent** spécifie la position de départ de la première ligne du paragraphe. Une valeur négative indique un "dépassement". La valeur maximale du "dépassement" est égale à celle spécifiée dans l'attribut **sidemargin**. Indiqué en multiple de 10 pt.

A.5.24.14 parskip

L'attribut **parskip** spécifie la largeur des espaces entre les paragraphes. Indiqué en multiple de 10 pt.

A.5.24.15 rubyalign

L'attribut **rubyalign** spécifie l'alignement du « Ruby Text ». Les valeurs d'alignement sont indiquées ci-dessous.

Tableau A.13 – Attribut rubyalign

Valeur d'alignement	
Aligné sur le début	Aligné sur le centre
start	center

A.5.24.16 rubyadjust

L'attribut **rubyadjust** spécifie le fonctionnement du « Ruby Text » au niveau du bord de la ligne. Les valeurs de l'opération sont indiquées ci-dessous. L'opération d'ajustement consiste à ajuster la position de début ou de fin du « Ruby Text » au niveau du bord de la ligne. Le saut de ligne fait référence à la méthode utilisée pour aligner le bord de début et le bord de fin lorsque le « Ruby Text » est plus long que la « Ruby Base ».

Tableau A.14 – Attribut rubyadjust

Valeur de l'opération de saut de ligne	
Opération utilisée	Opération non utilisée
line-edge	none

A.5.24.17 rubyoverhang

L'attribut **rubyoverhang** indique si l'opération de dépassement du « Ruby Text » est exécutée ou non, lorsque le « Ruby Text » est plus long que la « Ruby Base ». Les valeurs de l'opération sont indiquées ci-dessous.

Tableau A.15 – Attribut rubyoverhang

RubyOverhang	Valeur de dépassement
Dépassement	auto
Pas de dépassement	none

A.5.24.18 empdotsposition

L'attribut **empdotsposition** indique la position des points d'emphase. Les valeurs sont indiquées ci-dessous.

Tableau A.16 – Attribut empdotsposition

EmpDotsPosition	Valeur de la position
Avant (présentation verticale: droit, présentation horizontale: haut)	before
Après (présentation verticale: gauche, présentation horizontale: sous)	after

A.5.24.19 empdotscode

L'attribut **empdotscode** spécifie le code de caractère (ou le symbole) utilisé comme point d'emphase.

A.5.24.20 emplineposition

L'attribut **emplineposition** indique la position de la ligne d'emphase. Les valeurs sont indiquées ci-dessous.

Tableau A.17 – Attribut emplineposition

EmpLinePosition	Valeur de la position
Avant (présentation verticale: droit, présentation horizontale: haut)	before
Après (présentation verticale: gauche, présentation horizontale: sous)	after

A.5.24.21 emplinetype

L'attribut **emplinetype** indique le type de ligne d'emphase. Les valeurs sont indiquées ci-dessous.

Tableau A.18 – Attribut emplinetype

Type de ligne	Valeur en mode ligne
Aucun	none
Trait plein	solid
Pointillé	dotted
Tiret	dashed
Trait double	double

A.5.24.22 column

L'attribut **column** indique le nombre séquentiel de colonnes.

A.5.24.23 columnsep

L'attribut **columnsep** spécifie l'espace entre les colonnes. Indiqué en multiple de 10 pt.

A.5.24.24 align

L'attribut **align** spécifie le sens d'alignement du texte.

Tableau A.19 – Attribut align

Sens	Valeur
Tête	head
Centre	center
Pied	foot

A.5.24.25 textlinewidth

L'attribut **textlinewidth** spécifie la largeur de ligne pour les éléments **Empline** et **Box**. Indiqué en multiple de 10 pt.

A.5.24.26 linecolor

L'attribut **linecolor** spécifie la couleur de ligne pour les éléments **Empline** et **Box**.

A.5.24.27 textstyle

L'attribut **textstyle** spécifie "stylelabel" de l'élément **TextStyle** qui indique les informations nécessaires au rendu du texte.

A.5.24.28 fontweight

L'attribut **fontweight** spécifie l'épaisseur du texte. La valeur 400 est l'épaisseur normale, et 800 est "Bold" (gras).

A.5.24.29 objid

Voir A.5.22.19.

A.5.24.30 bgcolor

L'attribut **bgcolor** spécifie la couleur d'arrière-plan du Bloc.

A.5.24.31 bgimagemode

L'attribut **bgimagemode** précise la manière dont il convient de placer l'"Image" spécifiée dans l'attribut **refbgimage**.

La valeur est "fix": La position de l'image est à l'origine du Bloc.

La valeur est "tile": Les images occupent le « Bloc » à partir de l'origine.

La valeur est "center": La position de l'image est au centre du « Bloc ».

A.5.24.32 blockheight

L'attribut **blockheight** spécifie la hauteur du « Bloc ».

A.5.24.33 blockrule

L'attribut **blockrule** spécifie la manière de développer la taille du « Bloc » si le client élargit la taille des caractères sur les dispositifs de visualisation. Les valeurs de l'élément **TextBlock** sont spécifiées "horz-fixed", "horz-adjustable", "vert-fixed" et "vert-adjustable".

Tableau A.20 – Attribut blockrule

Valeur de la règle	Règle
horz-fixed	Impossible de modifier la largeur du bloc
horz-adjustable	La largeur du bloc peut être modifiée
vert-fixed	Impossible de modifier la hauteur du bloc
vert-adjustable	La hauteur du bloc peut être modifiée
block-fixed	Impossible de modifier la largeur et la hauteur du bloc
block-adjustable	La largeur et la hauteur du bloc peuvent être modifiées proportionnellement

A.5.24.34 blockstyle

L'attribut **blockstyle** spécifie "stylelabel" de l'élément **BlockStyle** qui indique les informations nécessaires au rendu du texte.

A.5.24.35 blockwidth

L'attribut **blockwidth** spécifie la largeur du « Bloc ».

A.5.24.36 footskip

L'attribut **footskip** définit le « saut de bas de page » de la « Composition de la mise en page ».

A.5.24.37 framecolor

L'attribut **framecolor** spécifie la couleur du cadre du « Bloc ».

A.5.24.38 framemode

L'attribut **framemode** spécifie le type de cadre du « Bloc ». Il est possible de spécifier deux types de cadre: "square" indique des angles droits aux quatre coins, et "curve" des coins arrondis.

A.5.24.39 framewidth

L'attribut **framewidth** spécifie la largeur du cadrage du « Bloc ». Si la valeur est "0" ou n'est pas indiquée, aucun cadre n'est rendu.

A.5.24.40 layout

L'attribut **layout** spécifie le système de coordonnées de l'élément **TextBlock**. Si "LrTb" est indiqué, l'origine se trouve en haut à gauche, l'abscisse et l'ordonnée étant respectivement orientées vers la droite et vers le bas. Si "TbRl" est indiqué, l'origine se trouve en haut à droite, l'abscisse et l'ordonnée étant respectivement orientées vers le bas et vers la gauche.

A.5.24.41 refbgimage

L'attribut **refbgimage** spécifie le "objid" de l'élément **Image** à afficher en arrière-plan de l'élément **TextBlock**. Aucune image ne s'affiche si "objid" n'est pas spécifié.

A.5.24.42 sidemargin

L'attribut **sidemargin** définit la « marge latérale » de la « Composition de la mise en page ».

A.5.24.43 toclabel

L'attribut **toclabel** spécifie la chaîne de caractères à afficher en tant que liste TOC sur les dispositifs de visualisation. Si "toclabel" est spécifié, l'Outil de création prend les informations de cette étiquette et définit cette étiquette dans les informations de l'élément **TOC** en mode automatique de création de table des matières.

A.5.24.44 topskip

L'attribut **topskip** spécifie le « saut supérieur » de la « Composition de la mise en page ».

A.5.24.45 empdotsfontname

L'attribut **empdotsfontname** spécifie le nom de la police du caractère (ou du symbole) utilisé comme points d'emphase.

A.5.24.46 refempdotsfont

L'attribut **refempdotsfont** spécifie le "objid" de l'élément **Font** utilisé comme points d'emphase. Si la valeur est 0, la police du texte principal est utilisée.

A.5.25 P

L'élément **P** spécifie l'unité minimale du document. Dans un roman, cela signifie « Paragraphe ». Cet élément **P** est traité comme la plus petite unité qui peut être spécifiée en tant qu'expression électronique.

Tableau A.21 – Attribut de P

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
refesound	0	Chaîne décimale	Indique le « objid » de l'élément eSound

L'attribut **refesound** spécifie le "objid" de l'élément **eSecond** utilisé comme élément sonore intégré. Si le son intégré est spécifié ici, il est lu au moment de l'affichage de l'élément **P**.

A.5.26 Plot

L'élément **Plot** spécifie l'image et le bouton insérés.

Tableau A.22 – Attribut de Plot

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
adjustment	"bottom"	Chaîne	Indique "center", "baseline", "top" ou "bottom"
refobj (obligatoire)		Chaîne d'ID objet	Indique le « objid » de l'élément Image ou Button
xsize (obligatoire)	(Entier non signé)	Chaîne décimale	[pt]*10
ysize (obligatoire)	(Entier non signé)	Chaîne décimale	[pt]*10

A.5.26.1 xsize

L'attribut **xsize** spécifie la dimension de l'abscisse de l'image ou du bouton inséré. Indiqué en multiple de 10 pt.

A.5.26.2 ysize

L'attribut **ysize** spécifie la dimension de l'ordonnée de l'image ou du bouton inséré. Indiqué en multiple de 10 pt.

A.5.26.3 refobj

L'attribut **refobj** indique le « objid » de l'élément (éléments **Image** ou **Button**) spécifié comme cible « inline ».

A.5.26.4 adjustment

L'attribut **adjustment** spécifie la position de l'image insérée ou du bouton inséré sur la ligne de texte.

Dans le cas d'une présentation horizontale, la valeur "top" doit correspondre au bord supérieur de l'image et du bouton insérés, et de la ligne de texte. La valeur "center" doit correspondre à la ligne centrale entre les deux éléments. La valeur "baseline" est censée correspondre à la ligne de base des deux éléments. La valeur "bottom" est censée correspondre au bord inférieur des deux éléments.

Dans le cas d'une présentation verticale, la valeur "top" doit correspondre au bord droit de l'image et du bouton insérés, et de la ligne de texte. La valeur "center" doit correspondre à la ligne centrale des deux éléments. La valeur "baseline" est censée correspondre à la ligne de base des deux éléments. La valeur "bottom" est censée correspondre au bord gauche des deux éléments.

A.5.27 CR

L'élément **CR** indique un saut de ligne obligatoire. S'il est placé immédiatement après l'élément **P**, l'espace de ligne de base est la somme des attributs "baselineskip" et "parskip".

A.5.28 Fill

L'élément **Fill** indique le caractère de début, dont la longueur est variable.

Tableau A.23 – Attribut de Fill

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
code (obligatoire)		Chaîne hexadécimale	Peut être spécifié entre "0x0020" et "0xffff"

L'attribut **code** indique le numéro de code de caractère (UTF-16) utilisé dans l'élément **Fill**.

A.5.29 Space

L'élément **Space** indique l'espacement et le crénage entre les caractères.

Tableau A.24 – Attribut de Space

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
xsize (obligatoire)	(Entier signé)	Chaîne décimale	[dot]

L'attribut **xsize** indique l'espace dans le sens de l'abscisse. Le crénage est réalisé lorsqu'une valeur négative est indiquée.

A.5.30 CharButton

L'élément **CharButton** indique la plage de la chaîne de caractères permettant de fournir la fonction du bouton.

Tableau A.25 – Attribut de CharButton

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
refobj (obligatoire)		Chaîne d'ID objet	Indique le « objid » de l'élément Button

L'attribut **refobj** indique le « objid » de l'élément (**Button**) spécifié comme cible du « Caractère Button ».

A.5.31 Rubi

L'élément **Rubi** indique le « Ruby Text » et le « Ruby Base ».

A.5.32 Oyamoji

L'élément **Oyamoji** indique la « Ruby Base ».

A.5.33 Gaiji

L'élément **Gaiji** indique le code (UTF-16) du caractère externe, et la chaîne de caractères alternative si le caractère externe ne peut pas être restitué.

Tableau A.26 – Attribut de Gaiji

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
code (obligatoire)		Chaîne hexadécimale	Peut être spécifié entre "0x0020" et "0xffff"
fontfacename (obligatoire)		Chaîne	Indique la police
refobj (obligatoire)		Chaîne d'ID objet	Indique le « objid » de l'élément Font

A.5.33.1 refobj

L'attribut **refobj** indique le « objid » de l'élément (**Font**) spécifié comme cible du « Caractère Gaiji ».

A.5.33.2 fontfacename

Voir A.5.24.5.

A.5.33.3 code

L'attribut **code** indique le code de caractère utilisé comme caractère externe.

A.5.34 AltString

L'élément **AltString** indique la chaîne de caractères alternative. Cette chaîne est utilisée lorsque le dispositif d'affichage ne peut pas restituer la chaîne de caractères d'origine.

A.5.35 Org

L'élément **Org** indique la chaîne de caractères d'origine que le client souhaite utiliser.

A.5.36 Alt

L'élément **Alt** indique la chaîne de caractères alternative à la place de l'élément **Org**.

A.5.37 Rubimoji

L'élément **Rubimoji** indique le "Ruby Text" du "Ruby Base" indiqué dans l'élément **Oyamoji**.

A.5.38 Box

L'élément **Box** indique la chaîne de caractères à encadrer d'un trait de boîte.

Tableau A.27 – Attribut de Box

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
linetype (obligatoire)		Chaîne	"solid", "dotted", "dashed" ou "double" est sélectionné

L'attribut **linetype** spécifie le type de trait de la boîte. Les valeurs sont indiquées ci-dessous.

Tableau A.28 – Attribut linetype

Type de ligne	Valeur en mode ligne
Trait plein	solid
Pontillé	dotted
Tiret	dashed
Trait double	double

A.5.39 EmpDots

L'élément **EmpDots** indique la chaîne de caractères qui est ajoutée aux points d'emphase.

Tableau A.29 – Attribut d'EmpDots

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
empdotscode		Chaîne hexadécimale	Indique le code de caractères utilisé comme point d'emphase. En outre, il est nécessaire de préciser en même temps les attributs empdotsfontname et refempdotsfont
empdotsposition		Chaîne	Indique "before" ou "after"
empdotsfontname		Chaîne	Indique le nom de la police utilisée comme points d'emphase. En outre, il est nécessaire de préciser en même temps les attributs empdotscode et refempdotsfont

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
refempdotsfont		Chaîne d'ID objet	Indique le « objid » de l'élément Font . En outre, il est nécessaire de préciser en même temps les attributs empdotscode et empdotsname

A.5.39.1 empdotsposition

Voir A.5.24.18.

A.5.39.2 empdotscode

Voir A.5.24.19.

A.5.39.3 empdotsfontname

Voir A.5.24.45.

A.5.39.4 refempdotsfont

Voir A.5.24.46.

A.5.40 Empline

L'élément **Empline** indique la chaîne de caractères qui est ajoutée à la ligne d'emphase.

Tableau A.30 – Attribut de Empline

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
emplinetype		Chaîne	Indique "none", "solid", "dotted", "dashed" ou "double"
emplineposition		Chaîne	Indique "before" ou "after"

A.5.40.1 emplineposition

Voir A.5.24.20.

A.5.40.2 emplinetype

Voir A.5.24.21.

A.5.41 Sub

L'élément **Sub** indique une chaîne de caractères en indice.

A.5.42 Sup

L'élément **Sup** indique une chaîne de caractères en exposant.

A.5.43 Yoko

L'élément **Yoko** indique la chaîne de caractères qui a modifié le sens de la ligne de base de l'horizontale à la verticale.

A.5.44 Tate

L'élément **Tate** indique la chaîne de caractères Occidentale qui a pivoté de 90° dans le sens anti-horaire dans la présentation verticale.

A.5.45 Nekase

L'élément **Nekase** indique la chaîne de caractères verticale qui a pivoté de 90° dans le sens horaire dans la présentation verticale.

A.5.46 NoBR

L'élément **NoBR** indique la chaîne de caractères dont la division n'est pas autorisée.

A.5.47 DrawChar

L'élément **DrawChar** indique le caractère de lettrine.

Tableau A.31 – Attribut de DrawChar

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
line (obligatoire)	(Entier non signé)	Chaîne décimale	Indique le nombre de lignes

L'attribut **line** indique le nombre de lignes du caractère spécifié comme lettrine.

A.5.48 Italic

L'élément **Italic** indique la chaîne de caractères en italique.

A.5.49 Bold

L'élément **Bold** indique la chaîne de caractères en gras.

A.5.50 ImageBlock

L'élément **ImageBlock** spécifie les informations de configuration de "Block" et "Image" afin d'afficher le Bloc à l'écran. Il est recommandé que l'attribut de l'élément **ImageBlock** décrive uniquement les informations différentes de celles de l'élément **BlockStyle** spécifiées par "blockstyle". La chaîne de caractères alternative est spécifiée s'il est impossible d'ouvrir le fichier image.

Tableau A.32 – Attribut de ImageBlock

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
x0 (obligatoire)	(Entier non signé)	Chaîne décimale	[dot]
y0 (obligatoire)	(Entier non signé)	Chaîne décimale	[dot]
x1 (obligatoire)	(Entier non signé)	Chaîne décimale	[dot]
y1 (obligatoire)	(Entier non signé)	Chaîne décimale	[dot]
xsize (obligatoire)	(Entier non signé)	Chaîne décimale	[dot]
ysize (obligatoire)	(Entier non signé)	Chaîne décimale	[dot]
refstream		Chaîne d'ID objet	Indique le « objid » de l'élément ImageStream . Seul "refstream" peut être sélectionné dans l'élément Image
objid (obligatoire)		Chaîne	Indique la seule chaîne de caractères dans le fichier
bgcolor		Chaîne COLORREF	

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
bgimagemode		Chaîne	Indique "fix", "tile" ou "centering". En outre, il est nécessaire de préciser en même temps l'attribut refbgimage
blockheight	(Entier non signé)	Chaîne décimale	[dot]
blockrule		Chaîne	"block-fixed" ou "block-adjustable" est spécifié pour l'attribut blockrule de ImageBlock
blockstyle (obligatoire)		Chaîne de style	Il est nécessaire d'associer la chaîne de caractères de style spécifiée à "stylelabel" de l'élément BlockStyle en référence
blockwidth	(Entier non signé)	Chaîne décimale	[dot]
footskip	(Entier non signé)	Chaîne décimale	[dot]
framecolor		Chaîne COLORREF	
framemode		Chaîne	Indique "curve" ou "square"
framewidth	(Entier non signé)	Chaîne décimale	[dot]
layout		Chaîne	Indique "LrTb" ou "TbRl"
refbgimage		Chaîne d'ID objet	Indique le « objid » de l'élément Image . Toutefois, l'ID propriétaire ne peut pas être spécifié
sidemargin	(Entier non signé)	Chaîne décimale	[dot]
topskip	(Entier non signé)	Chaîne décimale	[dot]

A.5.50.1 x0

L'attribut **x0** indique la coordonnée X du coin supérieur gauche nécessaire à l'extraction de la partie à afficher à partir de l'image source.

A.5.50.2 y0

L'attribut **y0** indique la coordonnée Y du coin supérieur gauche nécessaire à l'extraction de la partie à afficher à partir de l'image source.

A.5.50.3 x1

L'attribut **x1** indique la coordonnée X du coin inférieur droit nécessaire à l'extraction de la partie à afficher à partir de l'image source.

A.5.50.4 y1

L'attribut **y1** indique la coordonnée Y du coin inférieur droit nécessaire à l'extraction de la partie à afficher à partir de l'image source.

A.5.50.5 xsize

L'attribut **xsize** indique la taille de l'image extraite dans le sens de l'abscisse. Il s'agit d'une expansion le long de l'abscisse si la taille est supérieure à (x1-x0) et d'une contraction le long de l'abscisse si elle est inférieure à (x1-x0).

A.5.50.6 ysize

L'attribut **ysize** indique la taille de l'image extraite dans le sens de l'ordonnée. Il s'agit d'une expansion le long de l'ordonnée si la taille est supérieure à (y1-y0) et d'une contraction le long de l'ordonnée si elle est inférieure à (y1-y0).

A.5.50.7 refstream

L'attribut **refstream** spécifie le "objid" de l'élément **ImageStream** contenant les données de l'image source à afficher.

A.5.50.8 objid

Voir A.5.22.19.

A.5.50.9 bgcolor

Voir A.5.24.30.

A.5.50.10 bgimagemode

Voir A.5.24.31.

A.5.50.11 blockheight

Voir A.5.24.32.

A.5.50.12 blockrule

L'attribut **blockrule** spécifie la manière de développer la taille du « Bloc » si le client augmente la taille des caractères sur les dispositifs de visualisation. Les valeurs de l'élément **ImageBlock** sont "block-fixed" et "block-adjustable".

Concernant la table des valeurs, voir A.5.24.33.

A.5.50.13 blockstyle

Voir A.5.24.34.

A.5.50.14 blockwidth

Voir A.5.24.35.

A.5.50.15 footskip

Voir A.5.24.36.

A.5.50.16 framecolor

Voir A.5.24.37.

A.5.50.17 framemode

Voir A.5.24.38.

A.5.50.18 framewidth

Voir A.5.24.39.

A.5.50.19 layout

L'attribut **layout** spécifie le système de coordonnées de l'élément **ImageBlock**. Si "LrTb" est indiqué, l'origine se trouve en haut à gauche, l'abscisse et l'ordonnée étant respectivement orientées vers la droite et vers le bas. Si "TbRl" est indiqué, l'origine se trouve en haut à droite, l'abscisse et l'ordonnée étant respectivement orientées vers le bas et vers la gauche.

A.5.50.20 refbgimage

L'attribut **refbgimage** spécifie le "objid" de l'élément **Image** à afficher en arrière-plan de l'élément **ImageBlock**. Aucune image ne s'affiche si "objid" n'est pas spécifié.

A.5.50.21 sidemargin

Voir A.5.24.42.

A.5.50.22 topskip

Voir A.5.24.44.

A.5.51 ButtonBlock

L'élément **ButtonBlock** spécifie les informations de configuration de "Block" et "Text" afin d'afficher le Bloc à l'écran. Il est recommandé que l'attribut de l'élément **ButtonBlock** décrive uniquement les informations différentes de celles de l'élément **BlockStyle** spécifiées par "blockstyle". De même, l'élément **ButtonBlock** spécifie l'action réalisée lors de l'utilisation du bouton.

Le fonctionnement du bouton comporte 4 états: État de base (avant le choix) → Focalisation (choix) → Pression (lors de la pression) → Origine (après le relâchement du bouton). Ces états changent lors du fonctionnement du terminal.

Tableau A.33 – Attribut de ButtonBlock

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
objid (obligatoire)		Chaîne	Indique la seule chaîne de caractères dans le fichier
Bgcolor		Chaîne COLORREF	
bgimagemode		Chaîne	Indique "fix", "tile" ou "centering". En outre, il est nécessaire de préciser en même temps l'attribut refbgimage
blockheight	(Entier non signé)	Chaîne décimale	[dot]
blockrule		Chaîne	"block-fixed" ou "block-adjustable" est spécifié pour l'attribut blockrule de ButtonBlock
blockstyle (obligatoire)		Chaîne de style	Il est nécessaire d'associer la chaîne de caractères de style spécifiée à "stylelabel" de l'élément BlockStyle en référence
blockwidth	(Entier non signé)	Chaîne décimale	[dot]
footskip	(Entier non signé)	Chaîne décimale	[dot]
framecolor		Chaîne COLORREF	
framemode		Chaîne	Indique "curve" ou "square"
framewidth	(Entier non signé)	Chaîne décimale	[dot]
layout		Chaîne	Indique "LrTb" ou "TbRI"
refbgimage		Chaîne d'ID objet	Indique le « objid » de l'élément Image En outre, il est nécessaire de préciser en même temps l'attribut bgimagemode
sidemargin	(Entier non signé)	Chaîne décimale	[dot]
topskip	(Entier non signé)	Chaîne décimale	[dot]

A.5.51.1 objid

Voir A.5.22.19.

A.5.51.2 bgcolor

Voir A.5.24.30.

A.5.51.3 bgimagemode

Voir A.5.51.3.

A.5.51.4 blockheight

Voir A.5.24.32.

A.5.51.5 blockrule

L'attribut **blockrule** spécifie la manière de développer la taille du « Bloc » si le client augmente la taille des caractères sur les dispositifs de visualisation. Les valeurs de l'élément **ButtonBlock** sont "block-fixed" et "block-adjustable".

Concernant la table des valeurs, voir A.5.24.33.

A.5.51.6 blockstyle

Voir A.5.24.34.

A.5.51.7 blockwidth

Voir A.5.24.35.

A.5.51.8 footskip

Voir A.5.24.36.

A.5.51.9 framecolor

Voir A.5.24.37.

A.5.51.10 framemode

Voir A.5.24.38.

A.5.51.11 framewidth

Voir A.5.24.39.

A.5.51.12 layout

L'attribut **layout** spécifie le système de coordonnées de l'élément **ButtonBlock**. Si "LrTb" est indiqué, l'origine se trouve en haut à gauche, l'abscisse et l'ordonnée étant respectivement orientées vers la droite et vers le bas. Si "TbRl" est indiqué, l'origine se trouve en haut à droite, l'abscisse et l'ordonnée étant respectivement orientées vers le bas et vers la gauche.

A.5.51.13 refbgimage

L'attribut **refbgimage** spécifie le "objid" de l'élément **Image** à afficher en arrière-plan de l'élément **ButtonBlock**. Aucune image ne s'affiche si "objid" n'est pas spécifié.

A.5.51.14 sidemargin

Voir A.5.24.42.

A.5.51.15 topskip

Voir A.5.24.44.

A.5.52 BaseButton

L'élément **BaseButton** indique l'image du bouton à l'état normal.

Tableau A.34 – Attribut de BaseButton

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
refimage		Chaîne d'ID objet	Indique le « objid » de l'élément Image

L'attribut **refimage** spécifie le "objid" de l'élément **Image** qui s'affiche sous forme de bouton. Si le bouton est l'élément **CharButton**, une image ne s'affiche pas même si elle est spécifiée.

A.5.53 FocusinButton

L'élément **FocusinButton** spécifie l'image et l'action du bouton à l'état de focalisation. Il est possible de spécifier plusieurs opérations de bouton. Elles sont exécutées dans la séquence spécifiée.

Tableau A.35 – Attribut de FocusinButton

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
refimage		Chaîne d'ID objet	Indique le « objid » de l'élément Image

Voir A.5.52.

A.5.54 JumpTo

L'élément **JumpTo** spécifie le mouvement du bouton qui permet de sauter vers un certain élément.

Tableau A.36 – Attribut de JumpTo

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
refobj (obligatoire)		Chaîne d'ID objet	Indique le « objid » de l'élément TextBlock , ButtonBlock , ImageBlock ou Canvas .
refpage (obligatoire)		Chaîne d'ID objet	Indique le « objid » de l'élément Page

A.5.54.1 refpage

Voir A.5.20.2.

A.5.54.2 refobj

Voir A.5.20.1.

A.5.55 Run

L'élément **Run** spécifie les actions telles que « Window », « Popup window » et « Sound playback ».

Tableau A.37 – Attribut de Run

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
runoption (obligatoire)	"normal"	Chaîne	Indique "normal", "opposite", "center" ou "opposite-center"
refobj (obligatoire)		Chaîne d'ID objet	Indique le « objid » de l'élément PopUpWin , Window ou Sound

A.5.55.1 runoption

L'attribut **runoption** spécifie la position d'affichage de la Fenêtre et de la Fenêtre en incrustation de l'affichage à 2 écrans. Les valeurs sont indiquées ci-dessous.

Tableau A.38 – Attribut runoption

Position	Option de position
Écran latéral du bouton	normal
Écran latéral du bouton opposé	opposite
Centre de l'écran latéral du bouton	center
Centre de l'écran latéral du bouton opposé	opposite-center

A.5.55.2 refobj

L'attribut **refobj** indique le « objid » de l'élément (**Window**, **PopUpWin** ou **Sound**) spécifié comme cible de « Run action ».

A.5.56 SoundStop

L'élément **SoundStop** force l'arrêt de la lecture sonore asynchrone.

A.5.57 CloseWindow

L'élément **CloseWindow** force la fermeture de la fenêtre qui s'affiche.

A.5.58 PushButton

L'élément **PushButton** spécifie l'image et l'action du bouton à l'état bouton pressé. Il est possible de spécifier plusieurs opérations de bouton. Elles sont exécutées dans la séquence spécifiée.

Tableau A.39 – Attribut de PushButton

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
refimage		Chaîne d'ID objet	Indique le « objid » de l'élément Image

Voir A.5.52.

A.5.59 UpButton

L'élément **UpButton** spécifie l'image et l'action du bouton à l'état bouton relâché. Il est possible de spécifier plusieurs opérations de bouton. Elles sont exécutées dans la séquence spécifiée. L'élément **PopUpWin** ne peut pas être appelé par l'élément **Run**.

Tableau A.40 – Attribut de UpButton

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
refimage		Chaîne d'ID objet	Indique le « objid » de l'élément Image

Voir A.5.52.

A.5.60 Canvas

L'élément **Canvas** spécifie la zone de coordonnées absolues permettant de créer la présentation complexe.

Tableau A.41 – Attribut de Canvas

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
bgcolor	"0xff000000"	Chaîne COLORREF	
blockrule	block-fixed	Chaîne	Spécifie "block-fixed" ou "block-adjustable" comme valeur de l'attribut blockrule de l'élément Canvas
canvasheight	"0" (Entier non signé)	Chaîne décimale	[dot]
canvaswidth	"0" (Entier non signé)	Chaîne décimale	[dot]
framecolor	"0x00000000"	Chaîne COLORREF	
framewidth	"0" (Entier non signé)	Chaîne décimale	[dot]
layout	"LrTb"	Chaîne	Indique "LrTb" ou "TbRl"
objid (obligatoire)		Chaîne	Indique la seule chaîne de caractères dans le fichier
toclabel		Chaîne	Indique la chaîne de caractères utilisée dans la liste TOC
framemode	"square"	Chaîne	Indique "curve" ou "square"

A.5.60.1 canvaswidth

L'attribut **canvaswidth** spécifie la largeur de la zone de l'élément **Canvas**.

A.5.60.2 canvasheight

L'attribut **canvasheight** spécifie la hauteur de la zone de l'élément **Canvas**.

A.5.60.3 bgcolor

L'attribut **bgcolor** spécifie la couleur d'arrière-plan de la zone de l'élément **Canvas**.

A.5.60.4 layout

L'attribut **layout** spécifie le système de coordonnées de l'élément **Canvas**. Si "LrTb" est indiqué, l'origine se trouve en haut à gauche, l'abscisse et l'ordonnée étant respectivement orientées vers la droite et vers le bas. Si "TbRl" est indiqué, l'origine se trouve en haut à droite, l'abscisse et l'ordonnée étant respectivement orientées vers le bas et vers la gauche.

A.5.60.5 framewidth

L'attribut **framewidth** spécifie la largeur du cadrage de « Canvas ». Si la valeur est "0" ou n'est pas indiquée, aucun cadre n'est rendu.

A.5.60.6 framecolor

L'attribut **framecolor** spécifie la couleur du cadre de « Canvas ».

A.5.60.7 blockrule

L'attribut **blockrule** spécifie la manière de développer la taille du « Bloc » si le client augmente la taille des caractères sur les dispositifs de visualisation. Les valeurs de l'élément **Canvas** sont "block-fixed" et "block-adjustable".

Concernant la table des valeurs, voir A.5.24.33.

A.5.60.8 objid

Voir A.5.22.19.

A.5.60.9 toclabel

Toclabel permet d'attribuer un nom d'élément à "Canvas" pour l'utiliser dans la table des matières. L'attribut **toclabel** spécifie la chaîne de caractères à afficher en tant que liste TOC sur les dispositifs de visualisation. Si "toclabel" est spécifié, l'Outil de Création prend les informations de cette étiquette et définit cette étiquette dans les informations de l'élément **TOC** en mode automatique de création de table des matières.

A.5.60.10 framemode

L'attribut **framemode** spécifie le type de cadre de « Canvas ». Il est possible de spécifier deux types de cadre: "square" indique des angles droits aux quatre coins, et "curve" des coins arrondis.

A.5.61 PutObj

L'élément **PutObj** spécifie la position des éléments sur l'élément **Canvas**.

Tableau A.42 – Attribut de PutObj

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
refobj (obligatoire)		Chaîne d'ID objet	Indique le « objid » de l'élément PopUpWin , TextBlock , ImageBlock , ButtonBlock ou Sound De plus, s'il est utilisé dans l'élément Header ou Footer , TextBlock et ImageBlock peuvent être spécifiés.
x1 (obligatoire)	(Entier non signé)	Chaîne décimale	[dot]
y1 (obligatoire)	(Entier non signé)	Chaîne décimale	[dot]

A.5.61.1 x1

L'attribut **x1** spécifie les coordonnées d'abscisse permettant de déterminer le point d'origine de l'élément "objid" indiqué avec refobj.

A.5.61.2 y1

L'attribut **y1** spécifie les coordonnées d'ordonnée permettant de déterminer le point d'origine de l'élément "objid" indiqué avec refobj.

A.5.61.3 refobj

L'attribut **refobj** indique le « objid » de l'élément spécifié comme élément affiché.

A.5.62 Moveto

L'élément **Moveto** spécifie la position de départ suivante de **Lineto**, **DrawBox** et **DrawEllipse** dans l'élément **Canvas**.

Tableau A.43 – Attribut de Moveto

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
x1 (obligatoire)	(Entier signé)	Chaîne décimale	[dot]
y1 (obligatoire)	(Entier signé)	Chaîne décimale	[dot]

A.5.62.1 x1

L'attribut **x1** spécifie l'abscisse du point de départ du tracé.

A.5.62.2 y1

L'attribut **y1** spécifie l'ordonnée du point de départ du tracé.

A.5.63 Lineto

L'élément **Lineto** spécifie une ligne droite dans l'élément **Canvas**. La ligne droite est tracée entre le point de départ et la position (point final) spécifiée par l'élément **Lineto**. Une fois la ligne tracée, le point final indiqué devient le point de départ du tracé suivant.

Tableau A.44 – Attribut de Lineto

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
linecolor		Chaîne COLORREF	
linetype		Chaîne	Indique "solid", "dotted", "dashed", ou "double"
arrowtype		Chaîne	Indique "none", "begin", "end" ou "both"
linewidth	(Entier non signé)	Chaîne décimale	[dot]
x1 (obligatoire)	(Entier signé)	Chaîne décimale	[dot]
y1 (obligatoire)	(Entier signé)	Chaîne décimale	[dot]

A.5.63.1 x1

L'attribut **x1** spécifie l'abscisse du point final du tracé.

A.5.63.2 y1

L'attribut **y1** spécifie l'ordonnée du point final du tracé.

A.5.63.3 linewidth

L'attribut **linewidth** spécifie la largeur de la droite à tracer.

A.5.63.4 linecolor

L'attribut **linecolor** spécifie la couleur de la droite de l'élément **Lineto**.

A.5.63.5 linetype

L'attribut **linetype** spécifie le type de droite à tracer.

Concernant la table des valeurs, voir A.5.38.

A.5.63.6 arrowtype

L'attribut **arrowtype** indique si la droite tracée dans l'élément **Canvas** est une ligne droite ou une flèche.

Tableau A.45 – Attribut arrowtype

Type de flèche	Valeur
Ligne droite	none
Flèche de point de départ	begin
Flèche de point final	end
Flèches à deux extrémités	both

A.5.64 DrawBox

L'élément **DrawBox** spécifie un tracé rectangulaire dans l'élément **Canvas**. Le quadrilatère est tracé sous la forme d'une diagonale entre le point de départ et le point final spécifiés par l'élément **DrawBox**.

Tableau A.46 – Attribut de DrawBox

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
fillcolor		Chaîne COLORREF	
linecolor		Chaîne COLORREF	
linetype		Chaîne	Indique "solid", "dotted", "dashed", ou "double"
linewidth	(Entier non signé)	Chaîne décimale	[dot]
x1 (obligatoire)	(Entier signé)	Chaîne décimale	[dot]
y1 (obligatoire)	(Entier signé)	Chaîne décimale	[dot]

A.5.64.1 x1

Voir A.5.63.1.

A.5.64.2 y1

Voir A.5.63.2.

A.5.64.3 linewidth

L'attribut **linewidth** spécifie la largeur de la droite du rectangle à tracer.

A.5.64.4 linecolor

L'attribut **linecolor** spécifie la couleur de la droite de l'élément **DrawBox**.

A.5.64.5 linetype

L'attribut **linetype** spécifie le type de droite du rectangle à tracer. Les valeurs sont indiquées ci-dessous.

Concernant la table des valeurs, voir A.5.38.

A.5.64.6 fillcolor

L'attribut **fillcolor** détermine la couleur du corps du rectangle à tracer.

A.5.65 DrawEllipse

L'élément **DrawEllipse** spécifie un tracé elliptique dans l'élément **Canvas**. Le quadrilatère est tracé sous la forme d'une diagonale entre le point de départ et le point final spécifiés par l'élément **DrawEllipse**.

Tableau A.47 – Attribut de DrawEllipse

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
fillcolor		Chaîne COLORREF	
linecolor		Chaîne COLORREF	
linetype		chaîne	Indique "solid", "dotted", "dashed", ou "double"
linewidth	(Entier non signé)	Chaîne décimale	[dot]
x1 (obligatoire)	(Entier signé)	Chaîne décimale	[dot]
y1 (obligatoire)	(Entier signé)	Chaîne décimale	[dot]

A.5.65.1 x1

Voir A.5.63.1.

A.5.65.2 y1

Voir A.5.63.2.

A.5.65.3 linewidth

L'attribut **linewidth** spécifie la largeur de la droite de l'ellipse à tracer.

A.5.65.4 linecolor

L'attribut **linecolor** spécifie la couleur de la droite de l'élément **DrawEllipse**.

A.5.65.5 linetype

L'attribut **linetype** spécifie le type de droite de l'ellipse à tracer. Les valeurs sont indiquées ci-dessous.

Concernant la table des valeurs, voir A.5.38.

A.5.65.6 fillcolor

L'attribut **fillcolor** détermine la couleur du corps de l'ellipse à tracer.

A.5.66 RuledLine

L'élément **RuledLine** spécifie les lignes cotées.

Tableau A.48 – Attribut de RuledLine

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
linelength (obligatoire)	(Entier non signé)	Chaîne décimale	[dot]
linetype (obligatoire)		chaîne	Indique "solid", "dotted", "dashed", ou "double"
linewidth (obligatoire)	(Entier non signé)	Chaîne décimale	[dot]
linecolor	0x00000000	Chaîne COLORREF	

A.5.66.1 linewidth

L'attribut **linewidth** spécifie la largeur de la ligne cotée.

A.5.66.2 linelength

L'attribut **linelength** spécifie la longueur de la ligne cotée. Si la longueur n'est pas précisée, la ligne cotée est tracée sur toute la longueur de la zone de tracé disponible.

A.5.66.3 linetype

L'attribut **linetype** spécifie le type de ligne cotée. Les valeurs sont indiquées ci-dessous.

Concernant la table des valeurs, voir A.5.38.

A.5.66.4 linecolor

L'attribut **linecolor** spécifie la couleur de la droite de l'élément **RuledLine**.

A.5.67 Style

L'élément **Style** spécifie chaque type d'informations de style.

A.5.68 BookStyle

L'élément **BookStyle** indique les informations de contenu.

Tableau A.49 – Attribut de BookStyle

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
stylelabel (obligatoire)		Chaîne de style	Indique la chaîne de caractères de style
objid (obligatoire)		Chaîne	Indique la seule chaîne de caractères dans le fichier

A.5.68.1 stylelabel

L'attribut **stylelabel** spécifie un nom identifiable de l'élément **Style**.

A.5.68.2 objid

Voir A.5.22.19.

A.5.69 SetDefault

L'élément **SetDefault** spécifie la valeur par défaut des informations de présentation disponibles dans le contenu. La valeur définie ici est la valeur par défaut pour l'utilisation de chaque type d'objet.

Tableau A.50 – Attribut de SetDefault

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
rubyalign (obligatoire)	"start"	Chaîne	Indique la chaîne de caractères "start" ou "center"
rubyadjust (obligatoire)	"none"	Chaîne	Indique la chaîne de caractères "line-edge" ou "none"
rubyoverhang (obligatoire)	"none"	Chaîne	Indique la chaîne de caractères "auto" ou "none"
empdotsposition (obligatoire)	"before"	Chaîne	Indique "before" ou "after"
emlineposition (obligatoire)	"before"	Chaîne	Indique "before" ou "after"
emlinetype (obligatoire)	"none"	Chaîne hexadécimale	Indique la valeur en mode de ligne "none", "solid", "dotted", "dashed" ou "double"
empdotsfontname (obligatoire)		Chaîne	Indique le nom de la police utilisée pour les points d'emphase. En outre, il est nécessaire de préciser en même temps les attributs empdotscode et refempdotsfont
refempdotsfont (obligatoire)		Chaîne d'ID objet	Indique le « objid » de l'élément Font . En outre, il est nécessaire de préciser en même temps les attributs empdotscode et empdotsfontname
empdotscode (obligatoire)	0x3001	Chaîne hexadécimale	Indique le code de caractères utilisé comme point d'emphase. En outre, il est nécessaire de préciser en même temps les attributs empdotsfontname et refempdotsfont
setwaitpropt (obligatoire)	"noreplay"	Chaîne	Indique "replay" ou "noreplay"

A.5.69.1 rubyalign

Voir A.5.24.15.

A.5.69.2 rubyadjust

Voir A.5.24.16.

A.5.69.3 rubyoverhang

Voir A.5.24.17.

A.5.69.4 empdotsposition

Voir A.5.24.18.

A.5.69.5 empdotscode

Voir A.5.24.19.

A.5.69.6 emplineposition

Voir A.5.24.20.

A.5.69.7 emplinetype

Voir A.5.24.21.

A.5.69.8 empdotsfontname

Voir A.5.24.45.

A.5.69.9 refempdotsfont

Voir A.5.24.46.

A.5.69.10 setwaitprop

L'attribut **setwaitprop** spécifie le processus "Wait" dans l'élément **Page**. Le tableau ci-dessous permet de préciser si le processus est relancé.

Tableau A.51 – Attribut setwaitprop

Valeur de l'indicateur Wait	Spécification de l'opération
replay	Relance
noreplay	Pas de relance

A.5.70 RegistFont

L'élément **RegistFont** spécifie la police à ouvrir à l'avance.

Tableau A.52 – Attribut de RegistFont

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
fontfilename (obligatoire)		Chaîne	Indique le nom de fichier après ouverture
file (obligatoire)		Chaîne de chemin d'accès au fichier	Décrit le chemin d'accès absolu ou relatif accessible du fichier
encoding (obligatoire)		Chaîne	Indique "TTF", "OTF" ou "BF"
fontname (obligatoire)		Chaîne	Indique la police

A.5.70.1 fontfilename

L'attribut **fontfilename** spécifie le nom du fichier de police que le logiciel d'affichage doit utiliser à l'ouverture de la police.

A.5.70.2 file

L'attribut **file** spécifie le chemin d'accès et le nom du fichier de la police à intégrer.

A.5.70.3 fontname

L'attribut **fontname** spécifie le nom de la police à intégrer.

A.5.70.4 encoding

L'attribut **encoding** spécifie le format de données du fichier de police. Les chaînes de caractères qui peuvent être décrites sont "TTF", "OTF" et "BF". Toutefois, dans cette version, "OTF" et "BF" sont des mots réservés et ne peuvent pas être réellement définis.

A.5.71 BookSetting

L'élément **BookSetting** spécifie l'environnement supposé au moment de la création du contenu.

Tableau A.53 – Attribut de BookSetting

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
bindingdirection (obligatoire)		Chaîne	Indique "Lr" ou "Rl"
dpi (obligatoire)	(Entier non signé)	Chaîne décimale	[dpi]*10
screenheight (obligatoire)	(Entier non signé)	Chaîne décimale	[dot]
screenwidth (obligatoire)	(Entier non signé)	Chaîne décimale	[dot]
Colordepth	"24" (Entier non signé)	Chaîne décimale	Indique la profondeur de couleur, en bits

A.5.71.1 bindingdirection

L'attribut **bindingdirection** spécifie le sens de défilement de la page. Si "Lr" est spécifié, la page défile de gauche à droite. Si "Rl" est spécifié, la page défile de droite à gauche.

A.5.71.2 dpi

L'attribut **dpi** spécifie la valeur dpi supposée lorsque la présentation du contenu a été déterminée.

A.5.71.3 screenwidth

L'attribut **screenwidth** spécifie la largeur supposée de la zone d'affichage lorsque la présentation du contenu a été déterminée.

A.5.71.4 screenheight

L'attribut **screenheight** spécifie la hauteur supposée de la zone d'affichage lorsque la présentation du contenu a été déterminée.

A.5.71.5 colordepth

L'attribut **colordepth** spécifie les bits dont la profondeur de couleur a besoin pour afficher le contenu tel qu'il se présente.

A.5.72 TextStyle

L'élément **TextStyle** spécifie les informations "Text" disponibles dans le contenu.

Tableau A.54 – Attribut de TextStyle

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
objid (obligatoire)		Chaîne	Indique la seule chaîne de caractères dans le fichier
stylelabel (obligatoire)		Chaîne de style	Indique la chaîne de caractères de style
FontSize	"0" (Entier signé)	Chaîne décimale	[pt]*10
Fontwidth	"-10" (Entier signé)	Chaîne décimale	[pt]*10 Indique "-10" s'il ne modifie pas la forme de la police
Fontescapement	"0"	Chaîne décimale	Indique 0 ou 2 700
Fontorientation	"0"	Chaîne décimale	Indique 0 ou 2 700
fontfacename (obligatoire)		Chaîne	Indique la police
Textcolor	"0x00000000"	Chaîne COLORREF	
Textbgcolor	"0xff000000"	Chaîne COLORREF	
Wordspace	"0" (Entier signé)	Chaîne décimale	[pt]*10
Letterspace	"0" (Entier signé)	Chaîne décimale	[pt]*10
Baselineskip	"0" (Entier signé)	Chaîne décimale	[pt]*10
Linespace	"0" (Entier signé)	Chaîne décimale	[pt]*10
Parindent	"0" (Entier signé)	Chaîne décimale	[pt]*10
Parskip	"0" (Entier signé)	Chaîne décimale	[pt]*10
Rubyalign		Chaîne	Indique la chaîne de caractères "start" ou "center"
Rubyadjust		Chaîne	Indique la chaîne de caractères "line-edge" ou "none"
Rubyoverhang		Chaîne	Indique la chaîne de caractères "auto" ou "none"
Empdotsposition		Chaîne	Indique "before" ou "after"
Empdotscode		Chaîne	Indique le code de caractères utilisé comme point d'emphase. En outre, il est nécessaire de préciser en même temps les attributs empdotsfontname et refempdotsfont
Emplinesposition		Chaîne	Indique "before" ou "after"
Emplinetype		Chaîne	Indique la valeur en mode de ligne "none", "solid", "dotted", "dashed" ou "double"
Column	"1"	Chaîne décimale	Indique un entier compris entre 1 et 9
Columnsep	"0"	Chaîne décimale	[pt]*10
Align	"head"	Chaîne	Indique "head", "center" ou "foot"
Textlinewidth	"0" (Entier non signé)	Chaîne décimale	[pt]*10
Linecolor	"0x00000000"	Chaîne COLORREF	
Charspace	"0" (Entier signé)	Chaîne décimale	[pt]*10
Fontweight	"400"	Chaîne décimale	Indique une valeur comprise entre 1 et 1 000
empdotsfontname		Chaîne	Indique le nom de la police utilisée pour les points d'emphase. En outre, il est nécessaire de préciser en même temps les attributs

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
			empdotscode et refempdotsfont
Refempdotsfont		Chaîne d'ID objet	Indique le « objid » de l'élément Font . En outre, il est nécessaire de préciser en même temps les attributs empdotscode et empdotsfontname

A.5.72.1 objid

Voir A.5.22.19.

A.5.72.2 stylelabel

L'attribut **stylelabel** spécifie un nom identifiable de l'élément **Style**. Il est nécessaire de le mettre en correspondance avec l'attribut "textstyle" de l'élément **TextBlock**, de manière à y faire référence.

A.5.72.3 fontsize

Voir A.5.24.1.

A.5.72.4 fontwidth

Voir A.5.24.2.

A.5.72.5 fontescapement

Voir A.5.24.3.

A.5.72.6 fontorientation

Voir A.5.24.4.

A.5.72.7 fontfacename

Voir A.5.24.5.

A.5.72.8 textcolor

Voir A.5.24.6.

A.5.72.9 textbgcolor

Voir A.5.24.7.

A.5.72.10 wordspace

Voir A.5.24.8.

A.5.72.11 letterspace

Voir A.5.24.9.

A.5.72.12 baselineskip

Voir A.5.24.11.

A.5.72.13 linespace

Voir A.5.24.12.

A.5.72.14 parindent

Voir A.5.24.13.

A.5.72.15 parskip

Voir A.5.24.14.

A.5.72.16 rubyalign

Voir A.5.24.15.

A.5.72.17 rubyadjust

Voir A.5.24.16.

A.5.72.18 rubyoverhang

Voir A.5.24.17.

A.5.72.19 empdotsposition

Voir A.5.24.18.

A.5.72.20 empdotscode

Voir A.5.24.19.

A.5.72.21 emplineposition

Voir A.5.24.20.

A.5.72.22 emplinetype

Voir A.5.24.21.

A.5.72.23 column

Voir A.5.24.22.

A.5.72.24 columnsep

Voir A.5.24.23.

A.5.72.25 align

Voir A.5.24.24.

A.5.72.26 textlinewidth

Voir A.5.24.25.

A.5.72.27 linecolor

Voir A.5.24.26.

A.5.72.28 charspace

Voir A.5.24.10.

A.5.72.29 fontweight

Voir A.5.24.28.

A.5.72.30 empdotsfontname

Voir A.5.24.45.

A.5.72.31 refempdotsfont

Voir A.5.24.46.

A.5.73 BlockStyle

L'élément **BlockStyle** spécifie les informations "Block" disponibles dans le contenu.

Tableau A.55 – Attribut de BlockStyle

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
stylelabel (obligatoire)		Chaîne de style	Indique une chaîne de caractères de style
objid (obligatoire)		Chaîne	Indique la seule chaîne de caractères dans le fichier
blockwidth	"0" (Entier non signé)	Chaîne décimale	[dot]
blockheight	"0" (Entier non signé)	Chaîne décimale	[dot]
blockrule	"horz-adjustable"	Chaîne	Indique "horz-fixed", "horz-adjustable", "vert-fixed" ou "vert-adjustable" "block-fixed" ou "block-adjustable"
bgcolor	"0xff000000"	Chaîne COLORREF	
layout	"LrTb"	Chaîne	Permet de sélectionner "LrTb" ou "TbRl"
framewidth	"0" (Entier non signé)	Chaîne décimale	[dot]
framecolor	"0x00000000"	Chaîne COLORREF	
framemode	"square"	Chaîne	Indique "curve" ou "square"
topskip	"0" (Entier non signé)	Chaîne décimale	[dot]
sidemargin	"0" (Entier non signé)	Chaîne décimale	[dot]
footskip	"0" (Entier non signé)	Chaîne décimale	[dot]
refbgimage	"0"	Chaîne d'ID objet	Indique le « objid » de l'élément Image . En outre, il est nécessaire de préciser en même temps l'attribut bgimagemode
bgimagemode	"fix"	Chaîne	Indique "fix", "tile" ou "centering". En outre, il est nécessaire de préciser en même temps l'attribut refbgimage

A.5.73.1 stylelabel

L'attribut **stylelabel** spécifie un nom identifiable de l'élément **Style**. Il est nécessaire de le mettre en correspondance avec l'attribut "blockstyle" des éléments **TextBlock**, **ImageBlock** et **ButtonBlock** de manière à y faire référence.

A.5.73.2 objid

Voir A.5.22.19.

A.5.73.3 blockwidth

Voir A.5.24.35.

A.5.73.4 blockheight

Voir A.5.24.32.

A.5.73.5 blockrule

L'attribut **blockrule** spécifie la manière de développer la taille du « Bloc » si le client augmente la taille des caractères sur les dispositifs de visualisation. Les valeurs des éléments **TextBlock**, **ImageBlock** et **ButtonBlock** sont “block-fixed” et “block-adjustable”.

Concernant la table des valeurs, voir A.5.24.33.

A.5.73.6 bgcolor

Voir A.5.24.30.

A.5.73.7 layout

L'attribut **layout** spécifie le système de coordonnées des éléments **TextBlock**, **ImageBlock** et **ButtonBlock**. Si “LrTb” est indiqué, l'origine se trouve en haut à gauche, l'abscisse et l'ordonnée étant respectivement orientées vers la droite et vers le bas. Si “TbRl” est indiqué, l'origine se trouve en haut à droite, l'abscisse et l'ordonnée étant respectivement orientées vers le bas et vers la gauche.

A.5.73.8 framewidth

Voir A.5.24.39.

A.5.73.9 framecolor

Voir A.5.24.37.

A.5.73.10 framemode

Voir A.5.24.38.

A.5.73.11 topskip

Voir A.5.24.44.

A.5.73.12 sidemargin

Voir A.5.24.42.

A.5.73.13 footskip

Voir A.5.24.36.

A.5.73.14 refbgimage

L'attribut **refbgimage** spécifie le “objid” de l'élément **Image** à afficher en arrière-plan de l'élément **BlockStyle**. Aucune image ne s'affiche si “objid” n'est pas spécifié.

A.5.73.15 bgimagemode

Voir A.5.73.15.

A.5.74 PageStyle

L'élément **PageStyle** spécifie les informations "Page" disponibles dans le contenu.

Tableau A.56 – Attribut de PageStyle

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
stylelabel (obligatoire)		Chaîne de style	Indique une chaîne de caractères de style
objid (obligatoire)		Chaîne	Indique la seule chaîne de caractères dans le fichier
refbgimage	"0"	Chaîne d'ID objet	Indique le « objid » de l'élément Image . En outre, il est nécessaire de préciser en même temps l'attribut bgimagemode
Bgimagemode	"fix"	Chaîne	Indique "fix", "left" ou "centering". En outre, il est nécessaire de préciser en même temps l'attribut refbgimage
Evenfooterid	"0"	Chaîne d'ID objet	Indique le « objid » de l'élément Footer
Evenheaderid	"0"	Chaîne d'ID objet	Indique le « objid » de l'élément Header
Oddfooterid	"0"	Chaîne d'ID objet	Indique le « objid » de l'élément Footer
Oddheaderid	"0"	Chaîne d'ID objet	Indique le « objid » de l'élément Header
Pageposition	"any"	Chaîne	Indique "upper", "lower" ou "any"
Setemptyview	"show"	Chaîne	Indique "show" ou "empty"
Setwaitprop		Chaîne	Indique "replay" ou "noreplay"
Topmargin	"0" (Entier signé)	Chaîne décimale	[dot]
Headheight	"0" (Entier signé)	Chaîne décimale	[dot]
Headsep	"0" (Entier signé)	Chaîne décimale	[dot]
oddsidemargin	"0" (Entier non signé)	Chaîne décimale	[dot]
evensidemargin	"0" (Entier non signé)	Chaîne décimale	[dot]
Textheight	"0" (Entier non signé)	Chaîne décimale	[dot]
Textwidth	"0" (Entier non signé)	Chaîne décimale	[dot]
footheight	"0" (Entier signé)	Chaîne décimale	[dot]
Présentation	"LrTb"	Chaîne	Indique "LrTb" ou "TbRI"
Footspace	"0" (Entier signé)	Chaîne décimale	[dot]

A.5.74.1 stylelabel

L'attribut **stylelabel** spécifie un nom identifiable de l'élément **Style**. Il est nécessaire de le mettre en correspondance avec l'attribut "pagestyle" de l'élément **Page** de manière à y faire référence.

A.5.74.2 objid

Voir A.5.22.19.

A.5.74.3 refbgimage

L'attribut **refbgimage** spécifie le "objid" de l'élément **Image** à afficher en arrière-plan de l'élément **PageStyle**. Aucune image ne s'affiche si "objid" n'est pas spécifié.

A.5.74.4 bgimagemode

Voir A.5.22.8.

A.5.74.5 evenfooterid

Voir A.5.22.7.

A.5.74.6 evenheaderid

Voir A.5.22.5.

A.5.74.7 oddfooterid

Voir A.5.22.6.

A.5.74.8 oddheaderid

Voir A.5.22.4.

A.5.74.9 pageposition

Voir A.5.22.3.

A.5.74.10 setemptyview

L'attribut **setemptyview** indique s'il faut afficher la vue avec l'en-tête et le bas de page ou une vue complètement vide matérialisée par une page blanche à l'écran.

L'en-tête et le bas de page s'affichent lorsque "show" est spécifié. Une vue complètement vierge s'affiche si "empty" est spécifié.

A.5.74.11 setwaitprop

Voir A.5.69.10.

A.5.74.12 topmargin

Voir A.5.22.18.

A.5.74.13 headheight

Voir A.5.22.12.

A.5.74.14 headsep

Voir A.5.22.13.

A.5.74.15 oddsidemargin

Voir A.5.22.15.

A.5.74.16 evensidemargin

Voir A.5.22.9.

Il convient d'utiliser la valeur de l'attribut **evensidemargin**.

A.5.74.17 textheight

Voir A.5.22.16.

A.5.74.18 textwidth

Voir A.5.22.17.

A.5.74.19 footspace

Voir A.5.22.11.

A.5.74.20 footheight

Voir A.5.22.10.

A.5.74.21 layout

Voir A.22.14.

A.5.75 Objects

L'élément **Objects** spécifie les éléments utilisés de manière répétée et les éléments indépendants du flux de l'élément **Main** (les fenêtres, les sons, etc.). Dans les éléments "Objects", l'ordre de description de chaque élément est libre.

A.5.76 Window

L'élément **Window** spécifie la zone ("Window") d'une hiérarchie élevée à partir de l'écran sur le dispositif d'affichage principal du texte. Il est possible de composer la présentation du caractère et de l'image dans l'élément **Window**.

L'élément "Window" peut être éliminé en réalisant l'opération "close" (abordée dans l'élément "PopUpWindow" décrit ci-après).

Tableau A.57 – Attribut de Window

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
windowwidth	"0" (Entier non signé)	Chaîne décimale	[dot]
windowheight	"0" (Entier non signé)	Chaîne décimale	[dot]
Présentation	"LrTb"	Chaîne	Indique la chaîne de caractères "LrTb" ou "TbRl"
objid (obligatoire)		Chaîne	Indique la seule chaîne de caractères dans le fichier

A.5.76.1 windowwidth

L'attribut **windowwidth** spécifie la largeur de l'élément **Window**.

A.5.76.2 windowheight

L'attribut **windowheight** spécifie la hauteur de l'élément **Window**.

A.5.76.3 layout

L'attribut **layout** spécifie le système de coordonnées de l'élément **Window**. Si "LrTb" est indiqué, l'origine se trouve en haut à gauche, l'abscisse et l'ordonnée étant respectivement

orientées vers la droite et vers le bas. Si "TbRI" est indiqué, l'origine se trouve en haut à droite, l'abscisse et l'ordonnée étant respectivement orientées vers le bas et vers la gauche.

A.5.76.4 objid

Voir A.5.22.19.

A.5.77 PopUpWin

L'élément **PopUpWin** spécifie la zone ("PopUpWindow") d'une hiérarchie élevée à partir de l'écran sur le dispositif d'affichage principal du texte. Il est possible de décrire le caractère et l'image dans l'élément **PopUpWin**.

Tableau A.58 – Attribut de PopUpWin

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
objid (obligatoire)		Chaîne	Indique la seule chaîne de caractères dans le fichier
popupwinlabel (obligatoire)		Chaîne	

A.5.77.1 objid

Voir A.5.22.19.

A.5.78 Sound

L'élément **Sound** spécifie les informations des sons à reproduire.

Tableau A.59 – Attribut de Sound

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
times	"1"	Chaîne décimale	Indique le nombre de fois que les sons sont joués. "0" indique un nombre illimité de fois
playmode	"sync"	Chaîne	Indique "sync" ou "async"
refstream		Chaîne d'ID objet	Indique le « objid » de l'élément SoundStream . En outre, il est nécessaire de préciser en même temps l'attribut refstream
objid (obligatoire)		Chaîne	Indique la seule chaîne de caractères dans le fichier

A.5.78.1 times

L'attribut **times** indique le nombre de fois que les sons sont joués.

A.5.78.2 playmode

L'attribut **playmode** indique la méthode de reproduction des sons. Il indique "sync" pour une lecture synchrone ou "async" pour une lecture asynchrone.

A.5.78.3 refstream

L'attribut **refstream** spécifie le "objid" de l'élément **SoundStream** qui spécifie les données de la source sonore à reproduire.

A.5.78.4 objid

Voir A.5.22.19.

A.5.79 SoundStream

L'élément **SoundStream** spécifie les données source du son.

Tableau A.60 – Attribut de SoundStream

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
encoding (obligatoire)		Chaîne	Indique "PCM" ou "MP3"
file (obligatoire)		Chemin d'accès au fichier	Indique le chemin d'accès au fichier et le nom de fichier contenant les données
objid (obligatoire)		Chaîne	Indique la seule chaîne de caractères dans le fichier

A.5.79.1 encoding

L'attribut **encoding** spécifie le format de données du fichier de son. Les chaînes de caractères qui peuvent être décrites sont "PCM" et "MP3".

A.5.79.2 file

L'attribut **file** spécifie le chemin d'accès et le nom de fichier des données de son.

A.5.79.3 objid

Voir A.5.22.19.

A.5.80 ImageStream

L'élément **ImageStream** spécifie les données source de l'image.

Tableau A.61 – Attribut de ImageStream

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
encoding (obligatoire)		Chaîne	Indique "JPEG", "GIF", "BMP" ou "PNG"
file (obligatoire)		Chemin d'accès au fichier	Indique le chemin d'accès au fichier et le nom de fichier contenant les données
objid (obligatoire)		Chaîne	Indique la seule chaîne de caractères dans le fichier

A.5.80.1 encoding

L'attribut **encoding** spécifie le format de données du fichier image. Les chaînes de caractères qui peuvent être décrites sont "JPEG", "GIF", "BMP" et "PNG".

A.5.80.2 file

L'attribut **file** spécifie le chemin d'accès et le nom de fichier des données d'image.

A.5.80.3 objid

Voir A.5.22.19.

A.5.81 Header

L'élément **Header** spécifie les informations à afficher dans l'en-tête. La zone d'en-tête est déterminée par les attributs "textwidth" et "headheight" de l'élément **Page**.

Tableau A.62 – Attribut de Header

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
Présentation	"LrTb"	Chaîne	Indique "LrTb" ou "TbRl"
Bgcolor	"0xff00000"	Chaîne COLORREF	
Framewidth	"0" (Entier signé)	Chaîne décimale	[dot]
Framecolor	"0x00000000"	Chaîne COLORREF	
Frame mode	"square"		Indique "curve" ou "square"
objid (obligatoire)		Chaîne	Indique la seule chaîne de caractères dans le fichier

A.5.81.1 layout

L'attribut **layout** spécifie le système de coordonnées de l'élément **Header**. Si "LrTb" est indiqué, l'origine se trouve en haut à gauche, l'abscisse et l'ordonnée étant respectivement orientées vers la droite et vers le bas. Si "TbRl" est indiqué, l'origine se trouve en haut à droite, l'abscisse et l'ordonnée étant respectivement orientées vers le bas et vers la gauche.

A.5.81.2 bgcolor

L'attribut **bgcolor** spécifie la couleur d'arrière-plan de l'en-tête.

A.5.81.3 framewidth

L'attribut **framewidth** spécifie la largeur du cadrage de l'en-tête. Si la valeur est "0" ou n'est pas indiquée, aucun cadre n'est rendu.

A.5.81.4 framecolor

L'attribut **framecolor** spécifie la couleur du cadre de l'en-tête.

A.5.81.5 framemode

L'attribut **framemode** spécifie le type de cadre qui entoure la zone d'en-tête. Il est possible de spécifier deux types de cadre: "square" indique des angles droits aux quatre coins, et "curve" des coins arrondis.

A.5.81.6 objid

Voir A.5.22.19.

A.5.82 Footer

L'élément **Footer** spécifie les informations à afficher dans le bas de page. La zone de bas de page est déterminée par les attributs "textwidth" et "footheight" de l'élément **Page**.

Tableau A.63 – Attribut de Footer

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
Présentation	"LrTb"	Chaîne	Indique "LrTb" ou "TbRI"
Bgcolor	"0xff000000"	Chaîne COLORREF	
Framewidth	"0" (Entier signé)	Chaîne décimale	[dot]
Framecolor	"0x00000000"	Chaîne COLORREF	
Frame mode	"square"	Chaîne	Indique "curve" ou "square"
objid (obligatoire)		Chaîne	Indique la seule chaîne de caractères dans le fichier

A.5.82.1 layout

L'attribut **layout** spécifie le système de coordonnées de l'élément **Footer**. Si "LrTb" est indiqué, l'origine se trouve en haut à gauche, l'abscisse et l'ordonnée étant respectivement orientées vers la droite et vers le bas. Si "TbRI" est indiqué, l'origine se trouve en haut à droite, l'abscisse et l'ordonnée étant respectivement orientées vers le bas et vers la gauche.

A.5.82.2 bgcolor

L'attribut **bgcolor** spécifie la couleur d'arrière-plan du bas de page.

A.5.82.3 framewidth

L'attribut **framewidth** spécifie la largeur du cadrage du bas de page. Si la valeur est "0" ou n'est pas indiquée, aucun cadre n'est rendu.

A.5.82.4 framecolor

L'attribut **framecolor** spécifie la couleur du cadre du bas de page.

A.5.82.5 framemode

L'attribut **framemode** spécifie le type de cadre qui entoure la zone de bas de page. Il est possible de spécifier deux types de cadre: "square" indique des angles droits aux quatre coins, et "curve" des coins arrondis.

A.5.82.6 objid

Voir A.5.22.19.

A.5.83 eSound

L'élément **eSound** spécifie les informations du son intégré.

Tableau A.64 – Attribut de eSound

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
objid (obligatoire)		Chaîne	Indique la seule chaîne de caractères dans le fichier
refstream (obligatoire)		Chaîne d'ID objet	Indique le « objid » de l'élément SoundStream

A.5.83.1 objid

Voir A.5.22.19.

A.5.83.2 refstream

Voir A.5.78.3.

A.5.84 Font

L'élément **Font** spécifie les informations de la police.

Tableau A.65 – Attribut de Font

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
file (obligatoire)		Chaîne	Indique le chemin d'accès au fichier et le nom de fichier contenant les données
fontfilename (obligatoire)		Chaîne	Indique le nom de fichier à utiliser lorsque le fichier de police est ouvert
fontname (obligatoire)		Chaîne	Indique le nom de police à utiliser
objid (obligatoire)		Chaîne	Indique la seule chaîne de caractères dans le fichier
encoding (obligatoire)		Chaîne	Indique "TTF", "OTF" ou "BF"

A.5.84.1 file

Voir A.5.70.2.

A.5.84.2 fontfilename

Voir A.5.70.1.

A.5.84.3 fontname

Voir A.5.70.3.

A.5.84.4 objid

Voir A.5.22.19.

A.5.84.5 encoding

Voir A.5.70.4.

A.5.85 Image

L'élément **Image** spécifie les informations de l'image. Les coordonnées de l'attribut utilisé par l'élément **Image** ne dépendent pas des différentes présentations. Elles sont calculées dans le système de coordonnées dans lequel il place tous les points de départ à gauche.

De plus, s'il convient que l'élément **Image** ne soit pas un fichier image que le dispositif de visualisation ne peut pas afficher, la chaîne de caractères de substitution est spécifiée pour un élément.

Tableau A.66 – Attribut de Image

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
x0 (obligatoire)	(Entier non signé)	Chaîne décimale	[dot]
y0 (obligatoire)	(Entier non signé)	Chaîne décimale	[dot]
x1 (obligatoire)	(Entier non signé)	Chaîne décimale	[dot]
y1 (obligatoire)	(Entier non signé)	Chaîne décimale	[dot]
xsize (obligatoire)	(Entier non signé)	Chaîne décimale	[dot]
ysize (obligatoire)	(Entier non signé)	Chaîne décimale	[dot]
Refstream		Chaîne d'ID objet	Indique le « objid » de l'élément ImageStream . Seul "refstream" peut être sélectionné dans l'élément Image
objid (obligatoire)		Chaîne	Indique la seule chaîne de caractères dans le fichier

A.5.85.1 x0

Voir A.5.50.1.

A.5.85.2 y0

Voir A.5.50.2.

A.5.85.3 x1

Voir A.5.50.3.

A.5.85.4 y1

Voir A.5.50.4.

A.5.85.5 xsize

Voir A.5.50.5.

A.5.85.6 ysize

Voir A.5.50.6.

A.5.85.7 refstream

Voir A.5.50.7.

A.5.85.8 objid

Voir A.5.22.19.

A.5.86 ButtonL'élément **Button** spécifie l'opération réalisée lors de l'utilisation du bouton.

Le fonctionnement du bouton comporte 4 états: État de base → Focalisation → Pression → Origine. Ces états changent lors du fonctionnement du terminal.

Tableau A.67 – Attribut de Button

Nom de l'attribut	Valeur par défaut	Type de données	Commentaires
Objid		Chaîne	Indique la seule chaîne de caractères dans le fichier

Concernant l'attribut **objid**, voir A.5.22.19.

A.6 Éléments réservés

Les éléments ci-dessous peuvent être utilisés aux fins décrites dans le présent article et doivent faire l'objet d'un consensus entre les parties. Sinon, ils ne doivent pas être utilisés.

A.6.1 ShowOrder

L'élément **ShowOrder** spécifie l'ordre d'affichage de chaque élément disposé dans l'élément **Page**. Le contrôle de son affichage ne modifie pas la présentation.

A.6.2 SetText

L'élément **SetText** spécifie la modification d'attributs du texte.

A.6.3 div

L'élément **div** spécifie la modification d'attributs de plusieurs éléments **P** à la fois.

A.6.4 SendMessage

L'élément **SendMessage** spécifie un message envoyé à une application.

A.6.5 SimpleTextBlock

L'élément **SimpleTextBlock** spécifie les informations de configuration de "Block" et "Text" afin d'afficher le Bloc à l'écran. SimpleTextBlock comporte les limites d'expression de TextBlock et doit être traité comme une séquence simple.

A.6.6 MoveObj

L'élément **MoveObj** spécifie l'élément déplacé du point de départ vers le point final dans le temps imparti.

A.6.7 ViewPoint

L'élément **ViewPoint** spécifie les coordonnées médianes de la zone à élargir par action de l'utilisateur.

A.6.8 MiniPage

L'élément **MiniPage** spécifie la zone d'expression d'une présentation plus complexe. Dans l'élément **Page**, les éléments "Block" ne peuvent pas se trouver dans le sens de chargement du caractère. A l'inverse, dans l'élément **MiniPage**, les éléments "Block" peuvent se trouver dans le sens de chargement du caractère.

A.6.9 PageDiv

L'élément **PageDiv** spécifie la division de la zone de texte sur l'élément **Page**.

A.6.10 BlockList

L'élément **BlockList** spécifie la liste des éléments "Block" indiquant le sens de défilement des caractères.

A.6.11 Text

L'élément **Text** spécifie les informations de configuration de "Text" afin d'afficher "Block" à l'écran.

A.6.12 SimpleText

L'élément **SimpleText** spécifie les informations de configuration de "Text" afin d'afficher "Block" à l'écran.

A.6.13 Locate

L'élément **Locate** spécifie l'emplacement relatif des éléments "Block" dans l'élément **MiniPage**.

A.6.14 Import

L'élément **Import** spécifie les objets et fichiers externes en référence.

A.6.15 ObjLink

L'élément **ObjLink** spécifie l'objet externe accédé. Le format de l'élément **ObjLink** dépend de l'attribut **accesstype**.

A.6.16 FileLink

L'élément **FileLink** spécifie les fichiers externes accédés. Le format de l'élément **FileLink** dépend de l'attribut **accesstype**.

A.6.17 Solo

L'élément **Solo** spécifie l'élément "Page" indépendant de l'élément **Main**. L'élément "Page" indépendant peut uniquement être utilisé par l'action "JumpTo", etc. à partir d'un élément de **Main**.

A.6.18 Pages

L'élément **Pages** spécifie le groupe de pages continu. La continuité des pages défilantes est assurée dans les mêmes "Pages", mais il n'existe aucune continuité entre d'autres "Pages".

A.6.19 Template

L'élément **Template** spécifie les informations utilisées comme styles et objets standard lors de la création de contenu.

A.6.20 TemplateSet

L'élément **TemplateSet** spécifie l'ensemble contenant les éléments **Style** et **Objects**. Un "objid" local est utilisé.

A.6.21 PlotText

L'élément **PlotText** spécifie le texte en ligne.

A.6.22 AutoSpacing

L'élément **AutoSpacing** spécifie la chaîne de caractères permettant d'égaliser l'espace.

A.6.23 Span

L'élément **Span** spécifie la partie du texte dans laquelle les attributs de texte doivent être modifiés.

A.6.24 Property

L'élément **Property** spécifie les informations demandées par les outils de création, etc. Il s'agit pour l'instant d'un élément réservé.

IECNORM.COM: Click to view the full PDF of IEC 62448:2009

Annexe B (normative)

Format XMDF XML

B.1 Généralités

La présente annexe spécifie le format XML XMDF mentionné dans l'Article 6.

Le format XML XMDF est un format générique d'échange de données de livre électronique multimédia, destiné aux préparateurs de données et aux éditeurs plutôt qu'aux lecteurs, l'accent étant mis sur les dispositifs mobiles (une plate-forme cible, par exemple). A l'instar de HTML, ce format ne divise pas le document en pages fixes, mais détermine la présentation en fonction de la taille d'écran du dispositif de visualisation, de la police utilisée, etc. Dans le présent document, ce type de contenu est appelé contenu flottant ou dynamique, par opposition au contenu paginé.

B.2 Présentation de la structure du format

Le contenu flottant est en général composé de plusieurs flux concaténés. La présente norme ne précise aucune exigence particulière quant au moyen de diviser le contenu flottant en flux individuels. Cette décision revient au préparateur de données, afin de s'adapter aux différents types de contenu. Par exemple, un journal peut comporter un flux par article, un roman par chapitre, et ainsi de suite. Il est également possible de ne pas diviser le contenu et de ne disposer que d'un seul flux. Toutefois, il convient de noter que les flux particulièrement volumineux, ou qu'un très grand nombre de flux, peut avoir un impact sur les performances d'exécution, selon la version spécifique du dispositif de visualisation utilisé, la mémoire disponible, etc.

La structure de l'arborescence XML du format est présentée dans la Figure B.1.

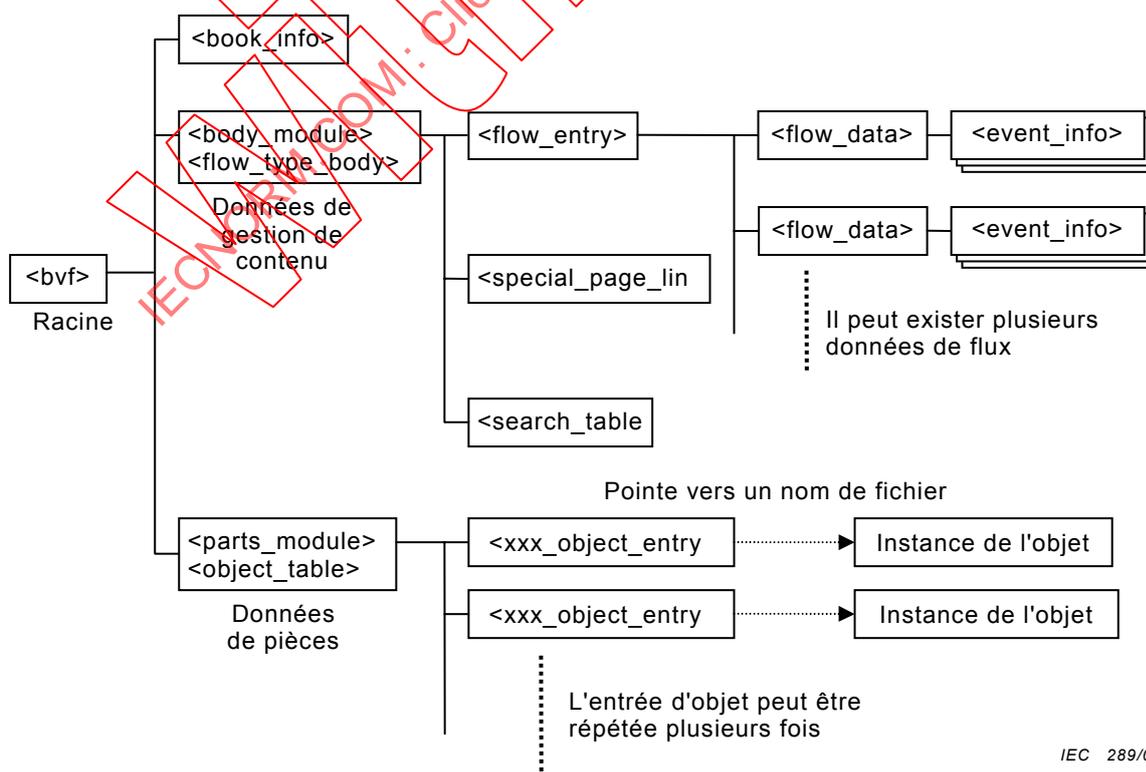


Figure B.1 – Structure de l'arborescence XML

En d'autres termes, le contenu réel de chaque flux que le dispositif de visualisation affiche, est enregistré dans l'*instance de l'objet*. L'*instance 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 spécification est générique et peut être utilisée dans tous les pays et toutes les langues. Toutefois, certaines parties peuvent générer un comportement spécifique à la langue. Pour raisonnablement simplifier la partie principale de la spécification et ne pas s'en écarter, les problèmes liés à la localisation sont présentés en détail en B.6. Le texte principal y fera référence en cas de besoin.

B.3 Éléments et attributs

B.3.1 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 font référence pour d'autres constructions. Dans les explications suivantes, les caractères alphanumériques font référence aux nombres compris entre 0 et 9 et les caractères alphabétiques sont compris entre a et z et A et Z.

B.3.2 Page_ID

Page_ID spécifie un numéro d'identification unique pour les données de flux du contenu flottant. Il s'agit d'une chaîne commençant par les caractères "PG" suivis de 4 caractères alphanumériques.

Exemple:

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

B.3.3 Object_ID

Object_ID spécifie un numéro d'identification unique pour les objets utilisés dans le contenu flottant. Il s'agit d'une chaîne commençant par les caractères "OB" suivis de 4 caractères alphanumériques.

Exemple:

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

B.3.4 Char_ID

Char_ID spécifie un numéro d'identification pour les chaînes de caractères à l'intérieur d'objets de texte. Il est unique dans le domaine d'application de l'*instance d'objet de texte* (voir B.4.6.2). Il s'agit d'une chaîne commençant par les caractères "CR" suivis de 4 caractères alphanumériques.

Exemple:

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

B.3.5 Lecture

Pour les besoins en matière de tri, la lecture peut être utile pour spécifier la lecture de chaque mot. Il est possible de faciliter la définition de la méthode de tri en restreignant les caractères admis à cette fin à un ensemble limité. Il convient de déterminer ces caractères par langue. Toutes les langues peuvent utiliser les caractères figurant dans le Tableau B.1 comme base commune, la localisation B.6 décrivant les extensions spécifiques à la langue.

Tableau B.1 – Caractères de base pour la lecture

Nom	Caractères correspondants (toutes les valeurs sont au format Unicode)
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)

Exemple:

```
<title reading="PI">π</title>
```

B.3.6 Nom de fichier

Il convient d'écrire les noms de fichier selon la convention ci-dessous. Les chemins d'accès varient en fonction du fichier dans lequel cette référence est faite. Il convient de ne pas utiliser de chemins d'accès 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"/>
```

B.3.7 Caractère standard

Le jeu de caractères standard du document, tel que défini par l'attribut `default_ccs` de la balise `<bvf>` (voir B.4.2), 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 XML XMDf. Toutes les données de livre électronique doivent définir leur jeu de caractères standard comme l'un (ou une combinaison) des jeux de caractères figurant dans le Tableau B.2 et ceux définis pour une localisation spécifique (voir B.6.2).

Tableau B.2 – Jeu de caractères standard

Nom du jeu de caractères	Description
"ISO/IEC 10646-IRV"	Caractères US-ASCII

B.3.8 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 ignoré.

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, il convient de remplacer les sauts de ligne (0x000D, 0x000A, 0x0D0A) et les tabulations (0x0009) en valeurs d'attribut par des espaces lors de la conversion au format de distribution.

B.3.9 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 de paires de substitution (0xD800 à 0xDFFF)

BOM (Byte Order Mark) (0xFFFE,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 les données d'un livre électronique utilisent un *Caractère étendu*, il convient d'ajouter le nom d'un jeu de caractères couvrant ces caractères supplémentaires 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.

B.3.10 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), les sauts de ligne (0x000D, 0x000A, 0x0D0A) et les tabulations (0x0009)) doivent être traités de la même manière que les *chaînes de caractères standard*.

B.3.11 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, 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 de 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 le 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 pointer vers le même caractère de chacun d'eux. Peut être ignoré.

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 *instance d'objet de texte*. Si `alt_img` et `alt_vimg` sont utilisés, les types de fichier doivent correspondre. Peut être ignoré. 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 de 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 ignoré, il convient d'utiliser l'image définie dans l'attribut `alt_img` pour la présentation horizontale et verticale. 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 *instance d'objet de texte*. Si `alt_img` et `alt_vimg` sont utilisés, les types de fichier doivent correspondre. Peut être ignoré. Si une valeur est affectée à cet attribut, il convient que l'affichage soit conforme aux mêmes méthodes que `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 *instance d'objet de texte*.

alt: Autre chaîne de caractères. Écrite comme une *chaîne de caractères standard*. Peut être ignoré.

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"/>
```

B.3.12 Chaîne de caractères externes

Une *chaîne de caractères standard* peut également contenir des *caractères externes*. Sauf indication contraire, il convient de traiter les caractères d'espacement (espace (0x0020), les sauts de ligne (0x000D, 0x000A) et les tabulations (0x0009)) de la même manière que dans les *chaînes de caractères standard*.

Exemple:

```
□ <external_char alt_img="ou.png" alt_type="image/png" alt="□"/>□
□□□ <external_char alt_set="sharp_extchars" alt_code="0x2345" alt="□"/>
```

B.3.13 Chaîne de caractères externes étendus

Une *chaîne de caractères standard* peut également contenir des *caractères étendus* et des *caractères externes*. Sauf indication contraire, il convient de traiter les caractères d'espacement (espace (0x0020), les sauts de ligne (0x000D, 0x000A) et les tabulations (0x0009)) de la même manière que dans les *chaînes de caractères standard*.

B.3.14 Coordonnées

Type de données permettant de stocker des coordonnées, les dimensions et d'autres informations analogues composées des valeurs x et 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'abscisse étant orientée vers la droite et l'ordonnée vers le bas. Étant donné que le système de coordonnées utilisé par le dispositif de visualisation dépend de la mise en oeuvre pour mettre des éléments en correspondance à l'écran, 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)"/>
```

B.3.15 Polygonal_region

Format de données permettant de stocker les sommets d'un polygone ou toute autre séquence ordonnée de sommets. Chaque sommet est stocké dans une balise <vertex>. Lors de la définition de la forme du polygone, il convient de ne pas croiser les arêtes. 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*. Ne peut pas être ignoré.

Exemple:

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

B.3.16 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 ignoré, il convient que le dispositif de visualisation agisse comme si RGB était défini.
- color:** Indique 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 B.6. 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 ignoré, sa valeur par défaut est 100.

Exemple:

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

B.3.17 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. 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>
```

B.3.18 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" >
```

B.3.19 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>
```

B.3.20 Personal_name

Format de données permettant de stocker des noms de personne (l'auteur du document, par exemple). Les noms sont placés dans la balise <personal_name>. Ses éléments enfants permettent de définir les différentes parties du nom: le prénom, le deuxième prénom et le nom de famille. Ces informations doivent être entrées par l'intermédiaire des balises définies ci-dessous. Noter que malgré leur caractère facultatif, l'un d'eux au moins doit être écrit.

[Éléments enfants]

<first_name> Définit le prénom. Écrit comme une *chaîne de caractères externes*. Peut être ignoré. Accepte l'attribut suivant:

[Attribut]

reading: prononciation du prénom, écrit comme un attribut *Reading*. Peut être ignoré.

<middle_name> Définit le prénom. Écrit comme une *chaîne de caractères externes*. Peut être ignoré. Accepte l'attribut suivant:

[Attribut]

reading: prononciation du deuxième prénom, écrit comme un attribut *Reading*. Peut être ignoré.

<last_name> Définit le nom de famille. Écrit comme une *chaîne de caractères externes*. Peut être ignoré. Accepte l'attribut suivant:

[Attribut]

reading: prononciation du nom de famille, écrit comme un attribut *Reading*. Peut être ignoré.

Exemple:

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

B.3.21 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 ignoré.

Exemple:

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

B.3.22 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 ignoré.
- <address> Permet de stocker l'adresse sous la forme d'une chaîne de caractères externes. Peut être ignoré.
- <telephone> Permet de stocker le numéro de téléphone sous la forme d'une *chaîne de caractères standard*. Peut être ignoré.
- <fax> Permet de stocker le numéro de télécopie sous la forme d'une *chaîne de caractères standard*. Peut être ignoré.
- <mail_address> Permet de stocker l'adresse électronique sous la forme d'une *chaîne de caractères standard*. Peut être ignoré.
- <website> Permet de stocker l'URI de la page d'accueil sous la forme d'une *chaîne de caractères standard*. Peut être ignoré.
- <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 ignoré. Si ces informations doivent être affichées, il convient d'afficher les espaces (0x0020) et les sauts de ligne (0x000D, 0x000A, 0x0D0A) en l'état, et 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>
```

B.3.23 Permission

Permet de définir les droits d'accès (les autorisations d'impression ou de copie, par exemple). Les différents droits d'accès sont stockés dans les éléments enfants de la balise <permission_info>. Si la balise <permission_info> est ignorée, la valeur attribuée lorsque chaque balise de droit d'accès est omise est affectée à tous les droits d'accès. Les explications ci-dessous font référence à une « authentification utilisateur ». Dans des circonstances habituelles, tous les dispositifs de visualisation 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> Indique si l'impression est autorisée ou interdite. S'il est ignoré, la valeur par défaut de l'attribut permission est « no ».

[Attribut]

permission: Indique 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> Indique si la copie est autorisée ou interdite. S'il est ignoré, la valeur par défaut de l'attribut permission est « no ».

[Attribut]

permission: Indique 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 permettant d'empêcher les copies externes, la copie est interdite.
- "no": personne ne peut copier (valeur par défaut).

Exemple:

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

B.3.24 Keyword

Avec la balise <keyword_list>, il est possible d'associer une liste de mots-clés aux données bibliographiques, à un flux 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: définit la catégorie du mot clé, écrite sous la forme d'une *chaîne de caractères standard*. Peut être ignoré.

reading: enregistre la prononciation du mot clé, écrite sous la forme d'un attribut *Reading*. Peut être ignoré.

[Élément enfant]

Chaîne de caractères externes: enregistre le mot clé actuel.

Exemple:

```

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

```

B.3.25 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 B.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. A savoir:

lorsque le numéro de téléphone est trop long pour le dispositif.

lorsque le numéro de téléphone contient des caractères ne figurant pas dans le Tableau B.3.

lorsque le numéro de téléphone contient une fonction qui ne peut pas être exécutée.

Tableau B.3 – Caractères utilisables d'un numéro de téléphone

Caractère	Code ASCII	Signification
0 à 9	0x30 à 0x39	numéro
#	0x23	bouton #
*	0x2A	bouton *
-	0x2D	Ignoré.
,	0x2C	Pause (1 seconde). Si une pause de 1 seconde ne peut être faite, 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.)

B.3.26 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 B.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 B.4 – Caractères utilisables des adresses électroniques

Catégorie	Caractères	ASCII code
nombres	0 à 9	0x30 à 0x39
alphabet	A à Z	0x41 à 0x5A
	a à z	0x61 à 0x7A
!	!	0x21
\$	\$	0x24

Catégorie	Caractères	ASCII code
%	%	0x25
&	&	0x26
*	*	0x2A
+	+	0x2B
-	-	0x2D
.	.	0x2E
/	/	0x2F
=	=	0x3D
?	?	0x3F
^	^	0x5E
_	_	0x5F
~	~	0x7E

B.4 Détails du format de description

B.4.1 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>
    ... <!-- Bibliography comes here -->
  </book_info>
  <body_module>
    ... <!-- Flow data comes here -->
  </body_module>
  <parts_module>
    ... <!-- Objects are registered here -->
  </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 B.6.8 pour plus de détails).

B.4.2 Modules d'informations du livre <bvf>

Le module d'information du livre est enregistré dans la balise <bvf>. Il fait office de balise de base 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:** Définit le type de numéro stocké dans l'attribut id. Écrit comme une *chaîne de caractères standard*. Peut être ignoré.
- 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 ignoré.
- default_ccs:** Permet de définir le nom du jeu de caractères des *caractères standard* et des *caractères étendus* utilisés dans le présent document (voir B.3). Si plusieurs jeux de caractères sont spécifiés, ils sont séparés par une virgule ("," - 0x002C). Ne peut pas être ignoré.
- 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 ignoré. Si le document comprend des dessins, cette taille représente également la taille de la scène.

[Éléments enfants]

- <book_info>** Permet d'enregistrer les données bibliographiques. Ne peut pas être ignoré. Voir B.4.2.1 pour plus de détails.
- <body_module>** Module de gestion du contenu. Ne peut pas être ignoré. Voir B.4.3 pour plus de détails.
- <parts_module>** Modules des données de pièces. Ne peut pas être ignoré. Voir B.4.5 pour plus de détails.

Exemple:

```

<bvf id_type="ISBN" id="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>

```

B.4.2.1 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. Ne peut pas être ignoré. 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 ignoré. L'attribut suivant peut être utilisé.

[Attribut]

- reading:** donne la prononciation du titre de la série sous la forme d'un attribut *Reading*. Peut être ignoré.

- <title>** Le titre est enregistré sous la forme d'une *chaîne de caractères externes* dans cet élément. Il ne peut pas être ignoré. L'attribut suivant peut être utilisé.

[Attribut]

reading: donne la prononciation du titre sous la forme d'un attribut *Reading*. Peut être ignoré.

<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 ignoré. L'attribut suivant peut être utilisé.

[Attribut]

reading: donne la prononciation du sous-titre sous la forme d'un attribut *Reading*. Peut être ignoré.

<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 ignoré. 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. 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 ignoré.

<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 ignoré. 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.

<author_info> Permet de stocker les informations relatives à l'auteur. Peut être ignoré. 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 ignoré, 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 ignoré.

<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 ignoré. 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, le caractère de tabulation (0x0009) devant l'être comme s'il s'agissait d'un espace.

<publisher_info> Permet de stocker les informations relatives à l'éditeur. Peut être ignoré. 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 ignoré. 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. Ne peut pas être ignoré. L'attribut suivant peut être utilisé.

[Attribut]

reading: donne la prononciation du nom de l'éditeur sous la forme d'un attribut *Reading*. Peut être ignoré.

<address_info> Permet de stocker l'adresse de l'éditeur, au format *Address* standard. Peut être ignoré.

<publisher_office> Permet de stocker des informations relatives à l'éditeur, s'il s'agit d'une entreprise. Peut être ignoré. Ses éléments enfants et attributs sont les suivants.

[Attribut]

publisher_code: Permet d'enregistrer l'ID de l'éditeur. Peut être ignoré.

[Éléments enfants]

<organization_name> Le nom de l'organisation est enregistré dans l'élément. Peut être ignoré.

<address_info> L'adresse de l'éditeur est enregistrée dans cet élément, au format *Address* standard. Peut être ignoré.

<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 ignoré. 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.

<seller_info> Permet de stocker les informations relatives au vendeur. Peut être ignoré. 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 ignoré. 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. Ne peut pas être ignoré. L'attribut suivant peut être utilisé.

[Attribut]

reading: donne la prononciation du nom du vendeur sous la forme d'un attribut *Reading*. Peut être ignoré.

<address_info> Permet de stocker l'adresse du vendeur, au format *Address* standard. Peut être ignoré.

<seller_office> Permet de stocker des informations relatives au vendeur, s'il s'agit d'une entreprise. Peut être ignoré. Ses éléments enfants et attributs sont les suivants.

[Attribut]

seller_code: Permet d'enregistrer l'ID du vendeur. Peut être ignoré.

[Éléments enfants]

<organization_name> Permet d'enregistrer le nom de l'organisation. Peut être ignoré.

<address_info> Permet d'enregistrer l'adresse du vendeur, écrite au format *Address* standard. Peut être ignoré.

<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 ignoré. 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.

<book_id_info> Permet d'enregistrer le numéro d'identification des livres (le numéro ISBN, par exemple). Peut être ignoré. Son élément enfant est le suivant.

[Élément enfant]

<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 ignoré, 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*. Ne peut pas être ignoré.

<classification_info> Permet de stocker des informations relatives à la classification du livre. Peut être ignoré. Son élément enfant est le suivant.

[Élément enfant]

<classification> Chaque type différent de classification est stocké dans une balise <classification> distincte. Si <classification_info> n'est pas ignoré, il doit exister au moins un <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. Ne peut pas être ignoré.

<rating> Permet de classer le contenu comme violent ou destiné aux adultes. Peut être ignoré. Les attributs suivants peuvent être utilisés.

[Attributs]

adult: Indique 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 ignoré.

<publication_date_info> Permet de stocker des informations relatives à la date de publication du livre. Peut être ignoré. Chaque date doit être stockée dans une instance distincte de l'élément enfant suivant.

[Élément enfant]

<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 ignoré, 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. S'il est ignoré, sa valeur par défaut est "publish".

<net_price_info> Permet de définir le prix du livre. Peut être ignoré, 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.

[Élément enfant]

<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 ignoré, 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 ignoré, le prix s'applique à tous les pays.

unit: permet de définir la devise sous la forme d'une *chaîne de caractères standard*. Ne peut pas être ignoré.

other_info: autres informations, écrites comme une *chaîne de caractères standard*. Peut être ignoré.

<book_abstract> Un résumé du livre, écrit sous la forme d'une *chaîne de caractères externes*, est enregistré dans cet élément. Peut être ignoré. 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 ignoré. 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 utilisé.

[Attribut]

type: permet de définir le type de l'image, en donnant son type MIME ("image/jpeg", par exemple). Ne peut pas être ignoré.

<spine_cover_image> Définit l'image à utiliser comme image spine. Suit les mêmes règles que <front_cover_image>. Peut être ignoré.

<keyword_list> Permet d'enregistrer une liste des mots clés relatifs aux données du livre, écrite dans la syntaxe *Keyword*. Peut être ignoré.

<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 ignoré. 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">4-1234-5678-9</book_id>
    <book_id type="Japaneze_ID_number">454745-7</book_id>
  </book_id_info>
  <classification_info>
    <classification type="Japaneze_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>

```

B.4.3 Module de gestion de contenu <body_module>

B.4.3.1 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

[Élément enfant]

<flow_type_body> Permet de traiter les données du contenu flottant. Voir B.4.3.2 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>

```

B.4.3.2 Données de contenu flottant <flow_type_body>

B.4.3.2.1 Balise <flow_type_body>

La balise <flow_type_body> traite les données du contenu flottant. Ses éléments enfants sont les suivants.

[Attribut]

Aucun

[Éléments enfants]

<flow_entry> Permet d'enregistrer les données de flux à utiliser comme contenu flottant du texte principal. Il ne doit exister qu'une seule instance de cet élément. Ne peut pas être ignoré. Voir B.4.3.2.2 pour plus de détails.

<special_page_link> Données de page spéciale. Permet de spécifier la position du contenu flottant de certaines pages souvent nécessaires (l'index ou le début du contenu principal, par exemple), afin de faciliter la référence. Peut être ignoré. Écrit comme indiqué en B.4.3.2.5. 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 ignoré. Écrit comme indiqué en B.4.3.2.6. L'omission de cet élément signifie qu'il n'existe aucune table de recherche.

B.4.3.2.2 Module d'enregistrement des données de flux <flow_entry>

La balise <flow_entry> permet d'enregistrer les données de flux à utiliser comme contenu flottant 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 les données de flux, telles que définies par l'élément <flow_data> suivant. Peut être ignoré. Voir B.4.3.2.3 pour plus de détails. S'il est ignoré, 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 flux. Il existe une relation univoque entre le nombre de balises <flow_data> et les données de flux réelles à enregistrer/afficher. Il doit exister une ou plusieurs instances de cette balise. L'ordre d'enregistrement des données de flux détermine l'ordre d'affichage. Voir B.4.3.2.4 pour plus de détails.

B.4.3.2.3 Module de l'attribut par défaut des données de flux <flow_default_attribute>

Permet de définir les valeurs par défaut des attributs à utiliser pour afficher les données de flux (voir B.4.3.2.4) dont est composé le contenu flottant principal. Les valeurs définies dans cette balise font office de valeurs par défaut pour tous les flux comportant un objet de texte, un objet d'écran de recherche ou un objet d'illustration comme données principales.

Une partie des valeurs qui peuvent être définies dans cette balise peut également être définie en local dans l'*instance d'objet de 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 *instances d'objet de texte* (voir B.4.6.2) 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> 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 oeuvre la méthode requise ou s'il souhaite donner la priorité aux paramètres utilisateur, il peut agir 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 ignorée, 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 flux. Les valeurs possibles sont indiquées ci-dessous. Peut être ignoré. 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 de texte*, voir B.4.6.2.2 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 flux. Les valeurs possibles sont indiquées ci-dessous. Peut être ignoré. Si cette valeur est ignorée, la valeur par défaut dépend du dispositif de visualisation.
- "portrait": Permet de choisir le mode portrait (plus grand que large). Toutefois, il peut être défini sur un autre sens selon le choix de l'utilisateur.
- "portrait_only": Permet de choisir le mode portrait (plus grand 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 grand). Toutefois, il peut être défini sur un autre sens selon le choix de l'utilisateur.
- "landscape_only": Permet de choisir le mode paysage (plus large que grand), 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.

[Éléments enfants]

<flow_default_size> Permet de définir l'interlettrage, la ligne de pente et la dimension de marge de tous les flux des données de contenu. Peut être ignoré. Si c'est le cas, il convient que le dispositif de visualisation se comporte comme si une valeur par défaut était affectée à tous ses attributs. Les attributs de cette balise sont les suivants.

[Attributs]

letter_spacing: Interlettrage par défaut. Les valeurs suivantes sont admises. Peut être ignoré. 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: Ligne de pente par défaut. Les valeurs suivantes sont admises. Peut être ignoré. 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 marge par défaut. Les valeurs suivantes sont admises. Peut être ignoré. 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 flux des données de contenu. Peut être ignoré. Si un flux individuel de données de contenu le définit également (avec la balise <text_default_font> de l'instance d'objet de texte, comme indiqué en B.4.6.2.2), les valeurs individuelles prévalent. S'ils sont tous deux ignorés, il convient que le comportement corresponde à la valeur par défaut des attributs 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 séparer chaque nom de police 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 ignoré. Si cet attribut et l'attribut du flux individuel sont ignorés, la valeur par défaut dépend du dispositif de visualisation.

fontsize: Taille de police par défaut. Les valeurs suivantes sont admises. Peut être ignoré. 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 3 possibilités dépendent du dispositif de visualisation et des capacités du dispositif sous-jacent.

bold_flag: Permet de choisir s'il convient d'afficher les données de contenu en gras ou non. Si cet attribut est ignoré, 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 instance d'objet de

texte, ceux de la balise dont la valeur "no" a été affectée à l'attribut bold_flag (voir B.4.6.2.3).

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 ignoré, la valeur dépend du dispositif de visualisation.

ruby_flag:

Permet de définir si le code ruby des données de contenu doit être visualisé ou non. S'il est ignoré, le comportement dépend du dispositif de visualisation. Le texte à afficher lorsque le code ruby est actif est celui placé dans la balise <ruby> des instances d'objet de *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 convient de ne pas afficher le code ruby, mais 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 de désactiver l'affichage du code ruby.

<flow_default_background> Permet de définir la couleur d'arrière-plan à utiliser pour tous les flux de données de contenu. Si un flux individuel de données de contenu le définit également (avec la balise <text_default_font> de l'instance d'objet de *texte*, comme indiqué en B.4.6.2.2), les valeurs individuelles prévalent. S'ils sont ignorés, il convient que le comportement corresponde à la valeur par défaut indiquée 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 ignoré, la valeur dépend du dispositif de visualisation.

<flow_default_line_breaking_method> Permet de spécifier l'algorithme à utiliser afin de déterminer la manière dont il convient de couper le texte en lignes. Les différentes langues traitent cela de diverses manières. Cet élément doit donc faire l'objet d'une certaine souplesse. Peut être ignoré. Si cette balise et les balises <line_breaking_method> de l'instance d'objet de *texte* (définies en B.4.6.2.2) 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. Les attributs et éléments enfants de cette balise sont les 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 contient des caractères, passer à la suivante.

D'autres valeurs sont possibles pour la localisation. Voir B.6.

D'autres attributs peuvent être ajoutés par des méthodes localisées. Voir B.6.

[Éléments enfants]

Aucun, à l'exception de ceux ajoutés par des méthodes localisées. Voir B.6.

B.4.3.2.4 Données de flux <flow_data>

Les données de flux sont définies ici. L'objet à utiliser comme contenu du flux 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 flux, écrit sous la forme d'un *Page_ID*. Peut être ignoré. <flow_entry> ne doit contenir aucune autre donnée de flux portant le même ID.

body_id: Pointe vers l'objet qui constitue le contenu de ce flux, utilisant l'*Object_ID* d'un objet "texte de contenu flottant", d'un objet "d'illustration" ou d'un objet "page de recherche". NE peut pas être ignoré.

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 flux précédent ou suivant en affectant l'une des valeurs ci-dessous à cet attribut. S'il est ignoré, sa valeur par défaut est "off". Toutefois, noter que lorsqu'un ID de corps pointe vers un objet de page de recherche, la valeur "on" doit être attribuée à turning_page_control.

"on": Tout déplacement vers les flux précédents et suivants est interdit.

"off": Les déplacements vers les flux précédents et suivants sont autorisés.

"forward": Le déplacement vers le flux suivant est interdit, et vers le flux précédent autorisé.

"back": Le déplacement vers le flux suivant est autorisé, et vers le flux précédent interdit.

Comme expliqué ci-dessus, ce paramètre permet de limiter les déplacements entre les flux, mais pas les déplacements à l'intérieur des flux. Pour limiter les déplacements à l'intérieur d'un objet de texte de contenu flottant, la commande de changement de page de la balise <page_break> (voir B.4.6.2.3) doit être utilisée.

[Élément enfant]

<event_info> Module d'informations d'événement. Permet de définir les événements (déclencheurs) et les actions associées. Peut être ignoré. Voir B.4.4 pour plus de détails. Si l'*Object_ID* défini dans le champ body_id fait référence à un objet d'illustration, cet élément event_info doit être ignoré.

B.4.3.2.5 Données de page spéciale <special_page_link>

Permet d'enregistrer la position des pages importantes ou 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 voyage, 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.

[Élément enfant]

<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 ignoré, 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 ignoré.

[Élément enfant]

Chaîne de caractères standard: Permet de spécifier la position à considérer comme une page spéciale en choisissant une donnée de flux particulière dans l'ensemble du contenu flottant. Ne peut pas être ignoré. Pour des données de flux textuelles, il est possible de spécifier une position à l'intérieur également.

- Spécifier uniquement les données de flux

Écrit comme un *Page_ID*. Il permet d'enregistrer l'ID de page des données de flux correspondantes.

Exemple:

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

- Spécifier les données de flux 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 flux. *Object_ID* est le numéro d'ID d'un objet de texte à l'intérieur de ce flux. *Char_ID* est le numéro d'ID d'une chaîne définie à l'intérieur de cet objet de texte.

Exemple:

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

Exemple: