BS EN 62448:2014

BSI Standards Publication

Multimedia systems and equipment — Multimedia e-publishing and e-books — Generic format for e-publishing

National foreword

This British Standard is the UK implementation of EN 62448:2014. It is identical to IEC 62448:2013. It supersedes BS EN 62448:2009 which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee EPL/100, Audio, video and multimedia systems and equipment.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Published by BSI Standards Limited 2014

ISBN 978 0 580 81620 8
ICS 33.160.60; 35.240.30

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 April 2014.

Amendments/corrigenda issued since publication

Date	Text affected

EUROPEAN STANDARD

NORME EUROPÉENNE

EUROPÄISCHE NORM

EN 62448

April 2014

ICS 33.160.60; 35.240.30

Supersedes EN 62448:2009

English version

Multimedia systems and equipment - Multimedia e-publishing and e-books - Generic format for e-publishing (IEC 62448:2013)

Systèmes et équipements multimédia - Publication et livres électroniques multimédia - Format générique pour la publication électronique (CEI 62448:2013)

Multimediasysteme und -geräte - E-Publishing und E-Books für Multimedia-Anwendungen - Allgemeines Format für E-Publishing (IEC 62448:2013)

This European Standard was approved by CENELEC on 2014-01-01. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

CENELEC

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

CEN-CENELEC Management Centre: Avenue Marnix 17, B - 1000 Brussels

Foreword

The text of document 100/2053/CDV, future edition 3 of IEC 62448, prepared by technical area 10, "Multimedia e- publishing and e-book technologies", of IEC/TC 100, "Audio, video and multimedia systems and equipment" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN 62448:2014.

The following dates are fixed:

• latest date by which the document has to be implemented at national level by publication of an identical national standard or by endorsement	(dop)	2014-10-18
• latest date by which the national standards conflicting with the document have to be withdrawn	(dow)	2017-01-01

This document supersedes EN 62448:2009.

EN 62448:2014 includes the following significant technical changes with respect to EN 62448:2009:

a) Addition of Annex C;

b) Related minor changes in Clause 6;

c) Updates in Annex B.

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

Endorsement notice

The text of the International Standard IEC 62448:2013 was approved by CENELEC as a European Standard without any modification.

In the official version, for Bibliography, the following notes have to be added for the standards indicated:

IEC 62605:2011	NOTE	Harmonised as EN 62605:2011 (not modified).

Annex ZA
(normative)

Normative references to international publications with their corresponding European publications

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

NOTE When an international publication has been modified by common modifications, indicated by (mod), the relevant EN/HD applies.

Publication	Year	Title	EN/HD	Year
ISO/IEC 19757-2	2008	Information technology - Document Schema Definition Language (DSDL) - Part 2: Regular-grammar-based validation - RELAX NG	-	-
ISO 639-1	-	Codes for the representation of names of languages - Part 1: Alpha-2 code	-	-
ISO 3166-1	-	Codes for the representation of names of countries and their subdivisions - Part 1: Country codes	-	-

CENELEC
European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

CEN-CENELEC Management Centre: Avenue Marnix 17, B - 1000 Brussels

CONTENTS

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.

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 either free of charge or 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 IEC. Information may be obtained from:

Sony Corporation
1-7-1 Konan, Minato-ku, Tokyo 108-0075, Japan

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

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
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. IEC shall not be held responsible for identifying any or all such patent rights.

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

MULTIMEDIA SYSTEMS AND EQUIPMENT – MULTIMEDIA E-PUBLISHING AND E-BOOKS – 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 documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 19757-2:2008*, Information technology – Document Schema Definition Language (DSDL) – Part 2: Regular-grammar-based validation – RELAX NG*

ISO 639-1, *Codes for the representation of names of languages – Part 1:Alpha-2 code*

ISO 3166-1, *Codes for the representation of names of countries and their subdivisions – Part 1: Country codes*

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

[SOURCE: IEC 62229:2006, 3.2]

3.2
data preparer
organization or person that prepares an e-book

Note 1 to entry: An editor is an example of a preparer.

[SOURCE: IEC 62229:2006, 3.3]

3.3
publisher
organization or person that issues and distributes an e-book

[SOURCE: IEC 62229:2006, 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 2788/13

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 19757-2:2008.

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 – BBeB Xylog, XMDF, and ESP formats. The structures of BBeB Xylog, XMDF, and ESP formats are shown in Annex A, Annex B, and Annex C, respectively.

```
default namespace="http://tc100.iec.ch/2005/ebook/generic"

start = ebook-g

ebook-g = ebook-g-core

        | external "bbebxylog.rnc"

        | external "xmdf.rnc"

        | external “esp.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 BBeB Xylog format mentioned in Clause 6.

This format is specified for two kinds of users: 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. The related tool developer can use it as a reference to specify the rules for the display equipment and to define the conversion rule between this standard and the other data forms such as any types of XML or HTML.

This annex includes the following features:

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 standard.

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.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 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.

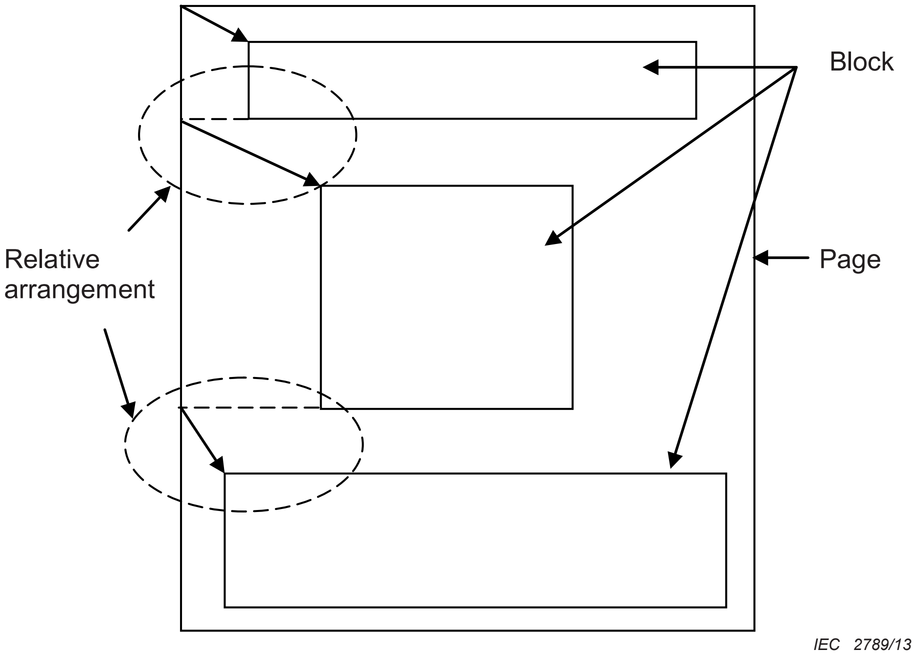


Figure A.1 – Conceptual diagram of the “Block layout”

A.3.2 Block

This standard 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.

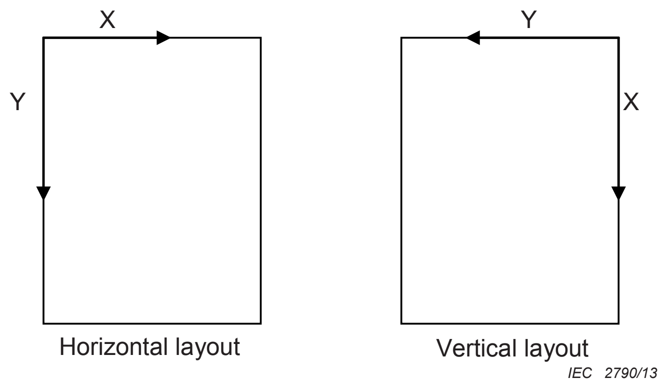


Figure A.2 – Coordinate system

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.

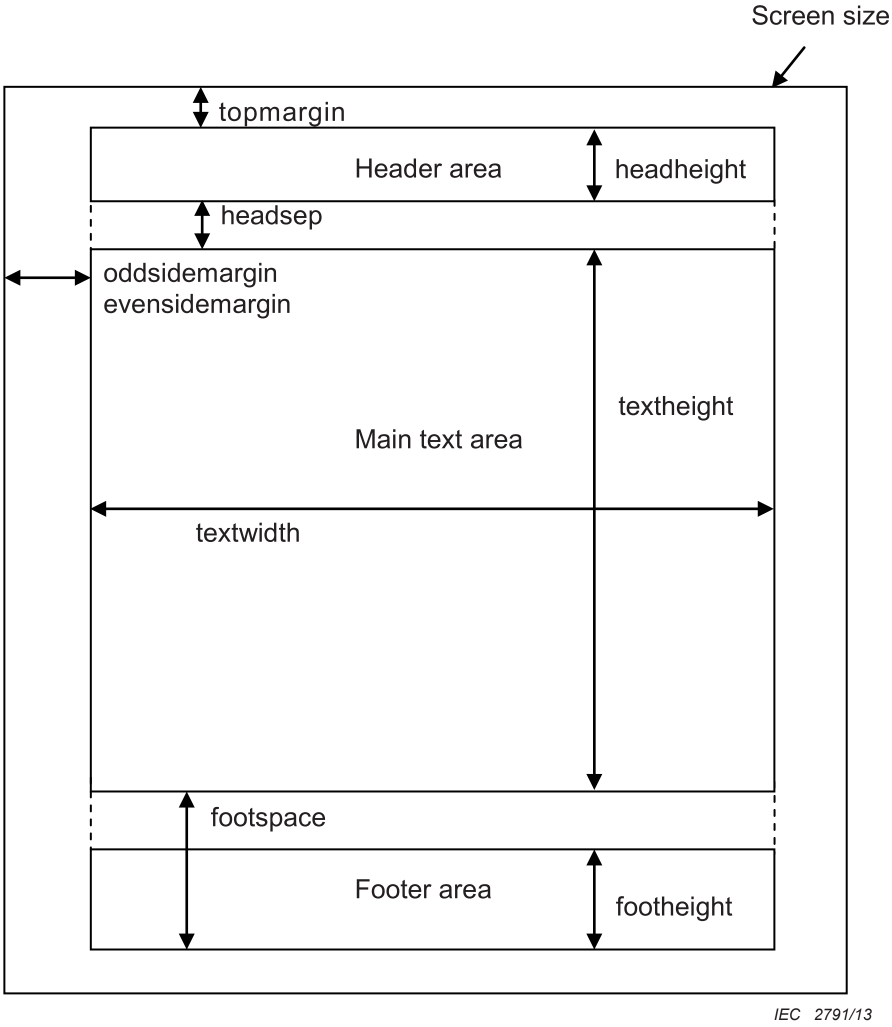


Figure A.3 – Page layout composition

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

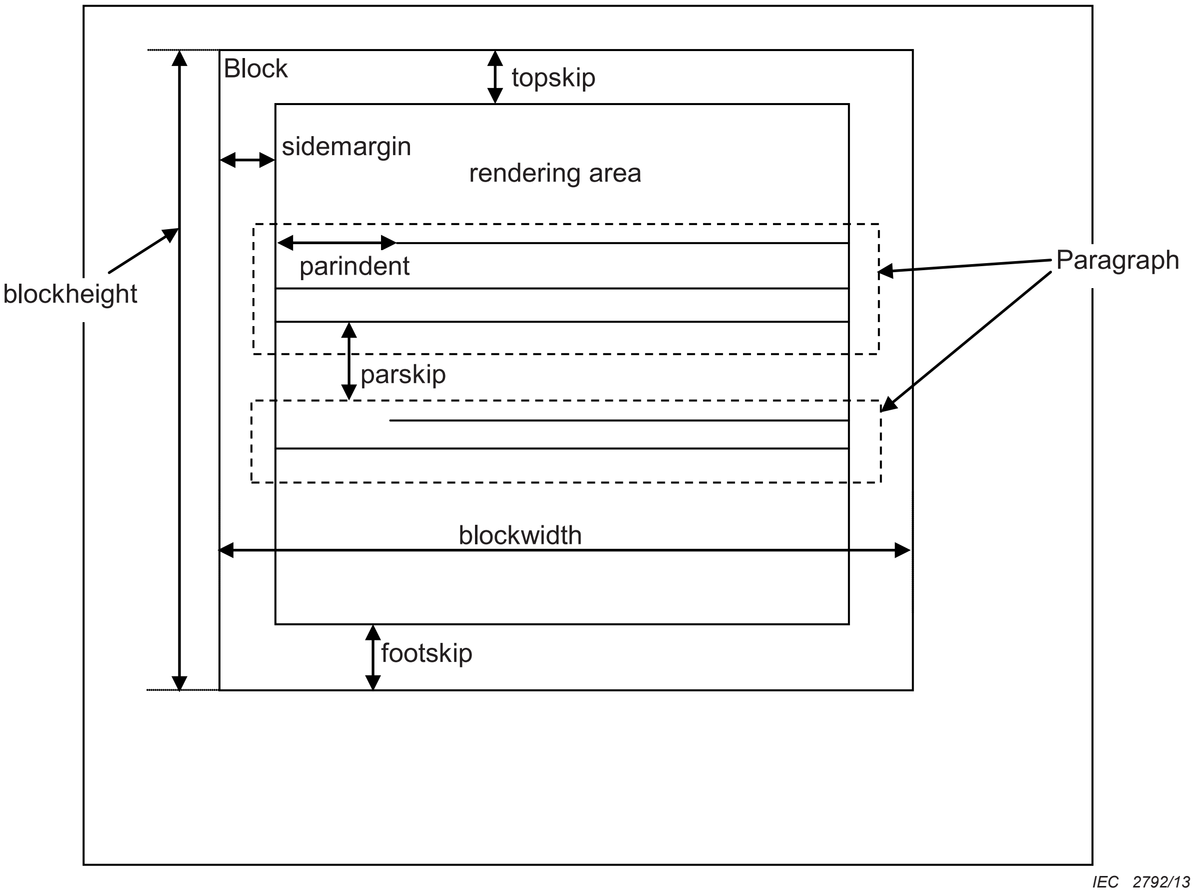


Figure A.4 – Block layout composition

A.4.3 Data types

The type definitions, as stated in Table A.1, 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 0xaarrggbb. (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", see Table A.2.

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, see Table A.3.

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, see Table A.4.

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 shall describe author names serially in an element, see Table A.5.

Table A.5 – Attribute of Author

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, see Table A.6.

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, see Table A.7.

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, see Table A.8.

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

A.5.20.0 General

The **TocLabel** element specifies the character string shown as the table of content list, see Table A.9.

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 specified as the "Jump" target.

A.5.21 Main

The **Main** element specifies the "Main content".

A.5.22 Page

A.5.22.0 General

The **Page** element specifies the elements composed of the "Page" and the 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", see Table A.10.

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 “TbRl”
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 **refgbimage** 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 “Rl” (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”.

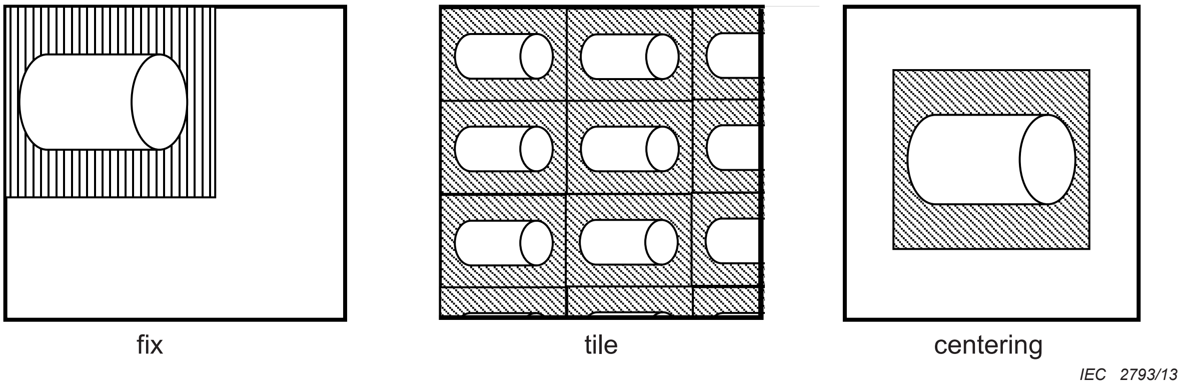


Figure A.5 – Bgimagemode attribute

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 footheight

The **footheight** attribute sets the “footheight” 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

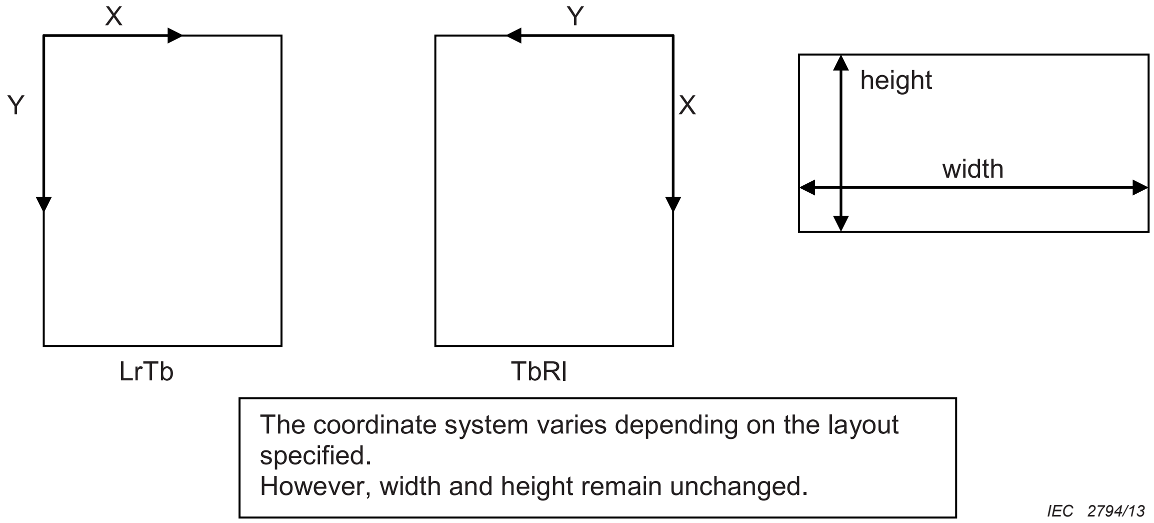
The **headheight** attribute sets the “headheight” of the “Page layout composition”.

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 “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.



The coordinate system varies depending on the layout specified.
However, width and height remain unchanged.

IEC 2794/13

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, see Table A.11.

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

A.5.24.0 General

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”, see Table A.12.

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 **refbgimage** 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”
emplineposition		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
parindent	(Signed integer)	Decimal string	[pt]*10

Attribute name	Default value	Data type	Comments
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 specifies 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, see Table A.13.

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”, see Table A.14.

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, see Table A.15.

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, see Table A.16.

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, see Table A.17.

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, see Table A.18.

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, see Table A.19.

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”, see Table A.20.

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 **refgbimage** 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, see Table A.21.

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

A.5.26.0 General

The **Plot** element specifies the inline image and button, see Table A.22.

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, see Table A.23.

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, see Table A.24.

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, see Table A.25.

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

A.5.33.0 General

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, see Table A.26.

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, see Table A.27.

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, see Table A.28.

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

A.5.39.0 General

The **EmpDots** element specifies the character string that is added the emphasis dots, see Table A.29.

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

A.5.40.0 General

The **Empline** element specifies the character string that is added to the emphasis line, see Table A.30.

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, see Table A.31.

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

A.5.50.0 General

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, see Table A.32.

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 **refbgimage** 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 “TbRl”
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 **refgbimage** 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

A.5.51.0 General

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) → Focus-in state(choice) → Push state(when push) → Up state(after release). These states are changed in turn by a user operation for the terminal, see Table A.33.

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 **refbgimage** 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”
refbgimage		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.5.24.31.

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 “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.5.51.13 refbgimage

The **refgbimage** 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, see Table A.34.

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, see Table A.35.

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

A.5.54.0 General

The **JumpTo** element specifies the movement of the button that jumps to a certain element, see Table A.36.

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

A.5.55.0 General

The **Run** element specifies the action such as “Window”, “Popup window” and “Sound playback” , see Table A.37.

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, see Table A.38.

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, see Table A.39.

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, see Table A.40.

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

A.5.60.0 General

The **Canvas** element specifies the absolute coordinates area to create the complex layout, see Table A.41.

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

A.5.61.0 General

The **PutObj** element specifies the position of elements on the **Canvas** element, see Table A.42.

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

A.5.62.0 General

The **Moveto** element specifies the next start position of the **Lineto**, **DrawBox** and **DrawEllipse** in the **Canvas** element, see Table A.43.

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

A.5.63.0 General

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, see Table A.44.

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, see Table A.45.

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

A.5.64.0 General

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, see Table A.46.

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

A.5.65.0 General

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, see Table A.47.

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

A.5.66.0 General

The **RuledLine** element specifies the ruled lines, see Table A.48.

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 ruled 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

A.5.68.0 General

The **BookStyle** element specifies the content information, see Table A.49.

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

A.5.69.0 General

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, see Table A.50.

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, see Table A.51.

Table A.51 – setwaitprop attribute

Wait flag value	Operation specification
replay	Replay
noreplay	No replay

A.5.70 RegistFont

A.5.70.0 General

The **RegistFont** element specifies the font to be opened in advance, see Table A.52.

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	Specifies 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

A.5.71.0 General

The **BookSetting** element specifies the assumed environment when the content was created, see Table A.53.

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
screeneheight **(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

A.5.72.0 General

The **TextStyle** element specifies the available “Text” information in the content, see Table A.54.

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
emplineposition		String	Specifies “before” or “after”
emplinetype		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	
charspace	“0” (Signed integer)	Decimal string	[pt]*10
fontweight	“400”	Decimal string	Specifies a value from "1" to "1,000"

Attribute name	Default value	Data type	Comments
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

A.5.73.0 General

The **BlockStyle** element specifies the available “Block” information in the content, see Table A.55.

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 “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.5.73.8 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 **refgbimage** 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.5.24.31.

A.5.74 PageStyle

A.5.74.0 General

The **PageStyle** element specifies the available “Page” information in the content, see Table A.56.

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 **Headter** 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]
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 **refgbimage** 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.5.22.14.

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

A.5.76.0 General

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, see Table A.57.

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 “TbRl” 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 “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.5.76.4 objid

Refer to A.5.22.19.

A.5.77 PopUpWin

A.5.77.0 General

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, see Table A.58.

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

A.5.78.0 General

The **Sound** element specifies the information of sounds to be playbacked, see Table A.59.

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 **SoundStrem** 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

A.5.79.0 General

The **SoundStream** element specifies the source data of the sound, see Table A.60.

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

A.5.80.0 General

The **ImageStream** element specifies the source data of the image, see Table A.61.

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

A.5.81.0 General

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, see Table A.62.

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

A.5.82.0 General

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, see Table A.63.

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

A.5.83.0 General

The **eSound** element specifies the information of the embedded sound, see Table A.64.

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

A.5.84.0 General

The **Font** element specifies the information of the font, see Table A.65.

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

A.5.85.0 General

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, see Table A.66.

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 → Focus-in state → Push state → Up state. These states are changed in turn by a user operation for the terminal, see Table A.67.

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

A.6.1 General

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.6.2 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.6.3 SetText

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

A.6.4 div

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

A.6.5 SendMessage

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

A.6.6 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.6.7 MoveObj

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

A.6.8 ViewPoint

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

A.6.9 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.6.10 PageDiv

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

A.6.11 BlockList

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

A.6.12 Text

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

A.6.13 SimpleText

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

A.6.14 Locate

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

A.6.15 Import

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

A.6.16 ObjLink

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

A.6.17 FileLink

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

A.6.18 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.6.19 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.6.20 Template

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

A.6.21 TemplateSet

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

A.6.22 PlotText

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

A.6.23 AutoSpacing

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

A.6.24 Span

The **Span** element specifies the particular portion of the text where the text attributes need to be changed.

A.6.25 Property

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

Annex B
(normative)

XMDF XML format

B.1 General

This annex describes the XMDF XML format mentioned in Clause 6.

XMDF 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 content 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 types of content. 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.

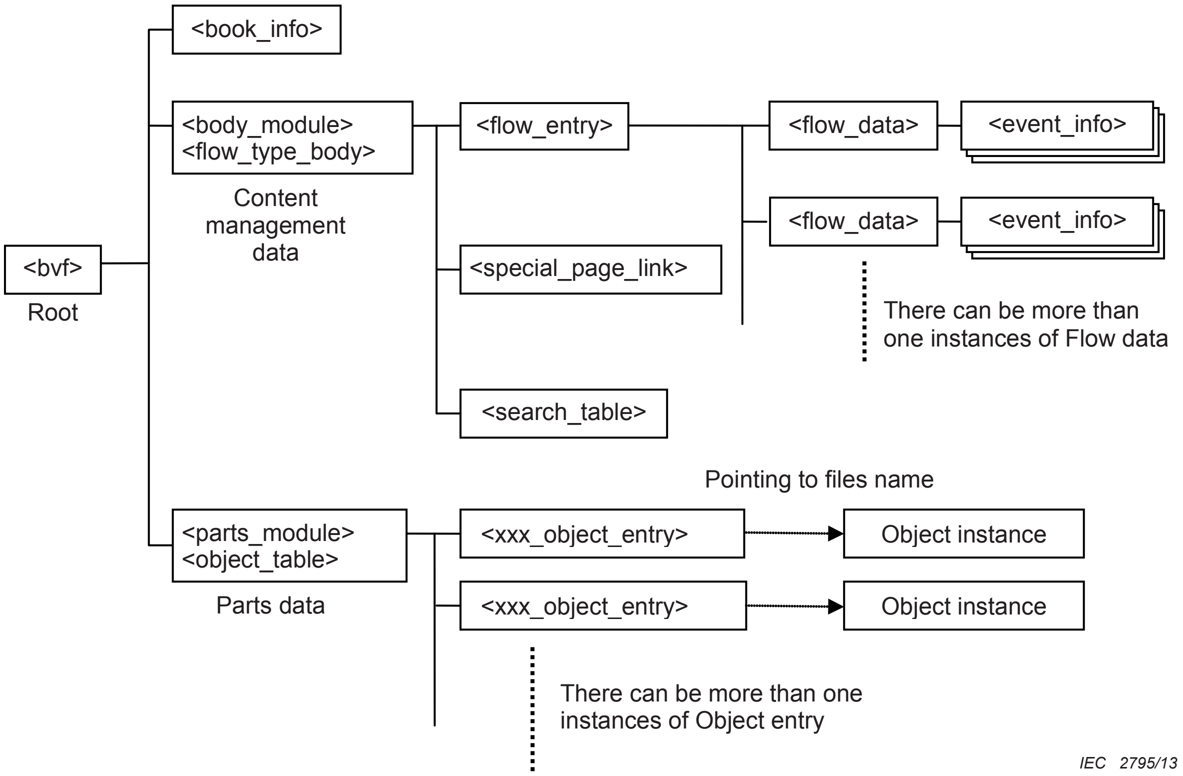


Figure B.1 – XML tree structure

The actual content 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 specifying such registered objects. In addition, information on functionalities such as page link is recorded in *event_info*.

The main part of this standard is generic, and can be used for any country and language. Some language-specific specifications are mainly described in Clause B.6 while 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 definitions detailed below will be valid throughout these specifications, 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, in half-width style.

B.3.2 Page_ID

A unique identification number for the flow data of the flowing content. It is a string starting with the "PG" characters, followed by 4 alphanumeric characters.

Example:

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

B.3.3 Object_ID

A unique identification number for objects used in the flowing content. It is a string starting with the "OB" characters, followed by 4 alphanumeric characters.

Example:

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

B.3.4 Char_ID

An identification number for character strings within text objects. It is unique in the scope of the *text object instance* (see B.4.7.2). It is a string starting with 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, it 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 in Clause 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. An absolute path or a relative path to the file in which this reference is made are used to specify. 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.

Table B.2 gives some examples of standard character sets.

Table B.2 – Examples of standard character sets

Character set name	Description	Remarks
"JIS X 0201,JIS X 0208:1997"	JIS1/JIS2 characrters.	
"ISO 646-IRV"	US-ASCII	
"ISO 8859-1"	ISO 8859-1(Latin-1)	
"ISO 8859-9"	ISO 8859-9(Latin-5)	
"ISO 8859-10"	ISO 8859-10(Latin-6)	
"ISO 8859-15"	ISO 8859-15(Latin-9)	
"JIS X 0201,JIS X 0208:1997"	(For Japanese) JIS1/JIS2 characters	
"JIS X 0201,X-SH-JIS 0213:2004"	(For Japanese) JIS1/2/3/4 characters except non-kanji JIS3/4 characters	It is recommended using UTF-8 encoding rather than Shift_JIS when this character set is specified by the default_ccs attribute of <bvf> tag.

B.3.8 Standard character string

A string composed of *Standard characters* is called a *Standard character string*. Unless otherwise specified, 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 character sets covering those extra characters should be appended to the *default_ccs* attribute of the <bvf> tag. Note that all *Extended characters* used in the document do not need to be covered by the same character set, as it is possible to specify several ones.

B.3.10 Extended character string

A string composed of *Standard characters* and *Extended characters* is called an *Extended character string*. Unless otherwise specified, 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 any one of the following methods.

a) Display the character set by the alt_set and alt_code attributes.

b) Display the image set by the alt_img or alt_vimg attributes.

c) Display the alternative letter set by the alt attribute

It has the following attributes.

[Attributes]

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

alt_set = "font1,font2, ..."

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

alt_code: Selects a character code point in the font specified by the alt_set attribute. It may be written both as a decimal number, or a hexadecimal number, prefixed by "0x". In case several fonts have been defined in the alt_set attribute, the character code shall 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* (including text object instance 2). When both alt_img and alt_vimg are used, the file types shall 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 <font> 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 <font> tag. The color of "gray" pixels shall be computed as an intermediate value between the font color and the background color.

c) Color images

Displayed as is

alt_vimg: Defines an alternative character image to be used when the text is displayed vertically (as can be the case in some languages, such as Japanese). Written as a *Filename*. When omitted, the image defined in the alt_img attribute should be used both for horizontal and vertical layout. Before opening the file indicated by this attribute, the img_type attribute should be checked for authorized file types. Note that it can be used only when <external_char> is used in a text object instance (including text object instance 2). When both alt_img and alt_vimg are used, the file types shall 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* (including text object instance 2).

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

Example:

```
<external_char alt_set="oooextchars" alt_code="47268" alt_img="ou.img"
               alt_vimg="ou_v.img" img_type="image/jpeg" alt="鴎"/>
<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 otherwise specified, 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="鴎'/>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 be used. Currently, only RGB is accepted. If this attribute is omitted, the viewer should act as if RGB was set.

color: Specifies the color name. Color names or numerical values 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 if color_space is set to/defaults to "RGB".

#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. Please refer to http://www.w3.org/TR/NOTE-datetime for details.

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 lowercase.

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 to define the various parts of the name: first name, middle name and last name. This information shall be entered via the tags defined below. Note that while they are all optional, you shall 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 middle 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>/(alternatively) <zipcode> 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> Defines whether printing is permitted or not. When omitted, it is to be considered as if the permission attribute was set to "no".

[Attribute]

permission: Defines whether printing is permitted or not. The following values are possible.

"authorized": authenticated users may print.

"no": no one can print (default value).

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

[Attribute]

permission: Defines whether copying is permitted or not. The following values are possible.

"authorized": authenticated users may copy.

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

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

Example:

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

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 standard, it is limited to bibliographical data). A <keyword_list> should contain one or more <keyword> tags as child element, each of these recording one keyword. The <keyword> tag has the following attributes and child element.

[Attributes]

category: Defines the category the keyword belongs to, written as a *Standard character string*. 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 shall be between 1 and 64, inclusive.

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

When the phone number string is too long for the device to handle.

When the phone number string includes characters not listed in Table 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 cannot be made, wait for key press instead.
/	0x2F	Pause (wait for key press).
P or p	0x50 or 0x70	Pause (wait for signal). If it is not possible to wait for the signal, wait for a key press instead.
+	0x2B	Sign to make an international call. Only at the beginning of a phone number. (If entered in another position, do not make the call.)

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) shall be written as an XML entity: "&amp;". 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 a to z	0x41 to 0x5A 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, while other encodings are not ruled out.

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.3.11). 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.4.3 for details.

<body_module> Content management module. Cannot be omitted. See B.4.4 for details.

<parts_module> Parts data modules. Cannot be omitted. See B.4.6 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.4.3 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 cannot 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> shall 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. Cannot 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> shall 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. Cannot 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 book's 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 shall be at least one <book_id>. The following attribute shall be set.

[Attribute]

type: Specifies the type of the identification number, such as "ISBN" for instance. Written as a *Standard character string*. 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 shall be at least one <classification>. It is stored as an *External character string*. The following attribute shall be set.

[Attribute]

type: type of the classification used. 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 shall 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 shall 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 shall be checked against the type attribute before opening the file. The following attribute shall be used.

[Attribute]

type: Defines the type of the image, by giving its MIME type, for example "image/jpeg". 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, Foobar 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>nonesense</keyword>
        </keyword_list>
</book_info>
```

B.4.4 Content management module <body_module>

B.4.4.1 <body_module> tag

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.4.4.2 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.4.4.2 Flowing content data <flow_type_body>

B.4.4.2.1 <flow_type_body> tag

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

[Attribute]

None

[Child elements]

<flow_entry> Registers the flow data to be used as the main text's flowing content. There shall only be one instance of this element. Cannot be omitted. See B.4.4.2.2 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.4.4.2.5. 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.4.4.2.6. Omission of this element means there is no search table.

B.4.4.2.2 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.4.4.2.3 for details. When omitted, the viewer should behave as if all of its attributes and child elements were set to their default value.

<flow_data> Registers information on each flow data. There is a one to one relation between the number of <flow_data> tags and actual flow data to be recorded/displayed. There shall be one or more instances of this tag. The order in which flow data are recorded determines the display order. See B.4.4.2.4 for details.

B.4.4.2.3 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.4.4.2.4), 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, a comic object or a dictionary data 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* and *dictionary data 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.4.7.2), <layout_default_attribute> tag of the *text object instance 2* and <dict_default_attribute> tag of the *dictionary data object instance* (see B.4.7.11) 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>, <layout_default_attribute>, and <dict_default_attribute> tags 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 as below. Can be omitted. If none of this attribute and its counterparts, i.e. the baseline attribute of the <text_default_attribute> tag in *text object instance*, (See B.4.7.2.2) and the baseline attribute of the <dict_default_attribute> tag in *dictionary data object instance* (See B.4.7.11.2), are 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 <layout_default_font> tag in *text object instance 2*, as defined in B.4.7.4.6.5, and <dict_default_font> tag in *dictionary data object instance*, as defined in B.4.7.11.1), the individual values take precedence. If the default font and the individual value are both 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 or values in unit of points or dots 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.

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 shall be displayed in bold style, except, in a *text object instance* and *dictionary data object instance*, those within a <font> tag with its “bold” attribute set to "no" (see B.4.7.2.3).

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 (with the options “yes” or “yes_only”) is the one sandwiched by the <ruby> tags in *text object instances* and *dictionary data object instances*. The following values can be used.

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

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

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

“no_only”: Ruby shall not be displayed, and cannot be turned on by the user. However, this does not apply if the viewer is incapable 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_background> tag of the *text object instance*, such as “color” attribute, as defined in B.4.7.2.2, or with the <dict_default_background> tag of the *dictionary data object*, as defined in B.4.7.11.2), the individual values take precedence. If all of these are absent, 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 background color of the display. Written in the standard *color* data format. If omitted, the value depends on the viewer.

<flow_default_line_breaking_method>/(alternatively) <flow_default_kinsoku> 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>/<kinsoku> tags in the *text object instance* (defined in B.4.7.2.1) and in the *dictionary data object instance* (defined in B.4.7.11.1) are specified, the latter takes priority. If all of these are absent, the default values of this <flow_default_line_breaking_method>/<flow_default_kinsoku> tag define the behavior. This tag has the following attributes and child elements.

[Attributes]

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

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

"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.

hanging_punctuation: Activates or deactivates the processing of hanging characters. Possible values are "yes" and "no". It 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.

[Child elements]

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

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

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

B.4.4.2.4 Flow Data <flow_data>

Flow data is defined in 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 shall be no other flow data with the same ID.

body_id: Specifies the object that will constitute this flow's content, using the *Object_ID* of a flowing content text object, of a comic object, of a "search page", or of a "dictionary data" 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 shall be set to "on".

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

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

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

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

As explained above, this setting limits moves from one flow to the others, but they do not restrict moves within each flow. To restrict moves inside a flowing content text object, use "turning_page_control" attribute of the <page_break> tag (see B.4.7.2.3), and to restrict moves inside a *dictionary data object*, use "turning_page_control" attribute of the <word> tag (see B.4.7.11.4).

head_button_control

Some viewers have keys for moving to the next/previous headword (i.e. the beginning of the definitions, see B.4.7.2.3 for details). However, it is possible to restrict such moves (from the end of this flow data to the next flow, or from the top of this flow data to the previous flow) by setting this attribute to one of the values listed below. When omitted, it defaults to "on". Note, however, that when body_id specifies a search page object, only "on" is the possible value. Setting is ignored when the viewer has no keys for such move, (which is often the case).

"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, use “head_button_control” attribute of the <page_break> tag (see B.4.7.2.3), and to restrict moves inside a *dictionary data object*, use “head_button_control" attribute of the <word> tag (see B.4.7.11.4).

[Child element]

<event_info> Event information module. Defines events (triggers) and associated actions. Can be omitted. See B.4.5 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.4.4.2.5 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 as below. It 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", ":afterward", "appendix", "answer", "glossary", "bibliography", "commentary", "index", "imprint", "author_info", "other", “flow_title”

Among the attributes above, if “flow_title” is set, this <special_page> tag is used as a tag for defining a character string to be displayed for the headline. Therefore, the viewer does not display this tag information in the list displayed in the table of contents menu. The viewer with the headline display function, however, displays the character string set for the “title” attribute in the headline.

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 or dictionary data (i.e., the object that is specified by the "body_id" attribute of the <flow_data> tag. See B.4.4.2.4 for details), it is possible to specify a position within it as well. If "flow_title" is set for the "kind" attribute, a flow to set character strings to be displayed for the headline is recorded according to conditions described in the "Specify only the flow data" section below. Also, if the <special_page> tag to specify "flow_title" for the "kind" attribute has the same flow for the child element (if numerous headline character strings are set for one flow) the viewer displays the first item written in the <special_page> tag as a headline character string.

– 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

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 or dictionary data object within that flow. *Char_ID* is the ID number of a string defined within that text object or dictionary data object.

Example:

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

Example:

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

B.4.4.2.6 Search table data <search_table>

Information defining the search tables is stored here, such as the search table's ID, or parameters concerning headwords registered in the search table. From here, the headword refers to the search's target. The headwords are registered according to the two different formats as stated below, 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 headword strings. In the distribution format, headwords are stored in this search table. The <search_table> tag has the following child elements.

Setting a headword

- Old format: Registers in the <key_entry> tag inside the text object entity. See for details about the <key_entry> tag.
- New format: Registers in the <key_entry> tag inside the dictionary data object entity. See B.4.7.11.4 for details.

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

[Attributes]

bookmark

Defines whether the search table contained in this content should be included in the viewer's bookmark feature for dictionary (a search history feature that records headwords automatically every time a search is executed). Accepts the values listed below. Can be omitted. When omitted, "no" is set.

"yes"	Included.
"no"	Not included.

wordbook

Defines whether the search table contained in this content should be included in the viewer's wordbook feature (a headwords registering and displaying feature). Accepts the values listed below. Can be omitted. When omitted, "no" is set.

"yes"	Included.
"no"	Not included.

jump_search_root

Defines whether this content should be a search source of the viewer's super jump search feature (a feature that searches through several contents using a character string selected from the body text as a search word from the content displayed status). Accepts the values listed below. Can be omitted. When omitted, "yes" is set.

"yes"	This content is selected as a search source.
"no"	This content is not selected as a search source.

jump_search

Defines whether the search table contained in this content should be included in the viewer's super jump search feature and W search feature (a feature that searches through several contents using a character string entered by a user as a search word from the content displayed status.) Accepts the values listed below. Can be omitted. When omitted, "yes" is set.

"yes"	Included.
"no"	Not included.

all_search

Defines whether the search table contained in this content should be included in the viewer's bundle search feature (a feature that searches through several contents using a character string entered by a user as a search word from the content non-displayed status). Accepts the values listed below. Can be omitted. When omitted, "yes" is set.

"yes"	Included.
"no"	Not included.

[Child element]

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

[Attributes]

id: Defines the ID number of this search table. The ID number shall be unique to each search table within the content. 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 headword entry fields (see B.4.7.9 for details) or from the headword information stored in *text object instance*'s <key_entry> tag (see B.4.7.2.3 for details).

use_default: Defines 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. It defaults to "no" when omitted.

"yes"	Can be used.
"no"	Cannot be used.

name: Defines the table name of this search table. Stored as a *Standard character string*. Up to 8 characters recommended. (Caution is given when the converter acknowledges the number of letters exceeds 8, with counting a full-length letter as one letter and a half-width letter as half a letter. If the number of letter exceeds 256, an error is given.) This table name is used for displaying the list of searched results, etc. Cannot be omitted.

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

"implicit"	Ordered as written in the content.
"unicode":	Ordered by Unicode value.
"other"	In a system-specific manner not expressed

short_name: Defines the short table name of this search table. Stored as a Standard character string. Up to 3 characters recommended. (Caution is given when the converter acknowledges the number of letters exceeds 3, with counting a two-byte character as one character and a single-byte character as half a character. If the number of letter exceeds 256, an error is given.) Cannot set a character string longer than the character string set in the "name" attribute. This short table name is used for displaying the list of searched results, etc. Cannot be omitted.

wild: Defines whether the viewer outputs a search table for a wild card search. Accepts the values listed below. Can be omitted. When omitted, "no" is set.

"yes"	Outputs.
"no"	Does not output.

blank: Defines whether the viewer outputs a search table for a blank word search. Accepts the values listed below. Can be omitted. When omitted, “no” is set.

"yes"	Outputs.
"no"	Does not output.

end: Defines whether the viewer outputs a search table for a word-ending search. Accepts the values listed below. Can be omitted. When omitted, “no” is set.

"yes"	Outputs.
"no"	Does not output.

The option for “wild”, “blank”, and “end” should be either all “yes” or all “no” in this standard.

help_page_id: Defines the page ID number for the flow data that writes explanations related to this search table in the *Pag_ID* format. Can be omitted.

[Child elements]

<enable_key_type> Defines the type of characters that can be used to store the lookup key of the search character strings in the search 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/(alternatively) number: Defines whether numbers should be used for the search character strings of the headwords. Accepts the values listed below. Can be omitted. When omitted, “no” is set.

"yes"	Allows use.
"no"	Does not allow use.

basic_alphabet/(alternatively) alphabet: Defines whether alphabets should be used for the search character strings of the headwords. Accepts the values listed below. Can be omitted. When omitted, “no” is set.

"yes"	Allows use.
"no"	Does not allow use.

kana Defines whether kana should be used for the search character strings of the headwords. Accepts the values listed below. Can be omitted. When omitted, “no” is set.

"yes"	Allows use.
"no"	Does not allow use.

kanji Defines whether kanji should be used for the search character strings of the headwords. Accepts the values listed below. Can be omitted. When omitted, “no” is set. However, the value should be only “no” if an old format is used for the keyword settings.

"yes"	Allows use.
"no"	Does not allow use.

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)
number	Half-width numerals: 0 to 9 (0x0030 to 0x0039) Full-width numerals: 0 to 9 (0xFF10 to 0xFF19)
alphabet	Standard US-ACSII alphabets: A to Z (0x0041 to 0x005A) and a to z (0x0061 to 0x007A) Full-width alphabets: A - Z (0xFF21 to 0xFF3A) a - z (0xFF41 to 0xFF5A)
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ﾚﾛﾜﾝﾞﾟ(0xFF71 to 0xFF9F), ｦｧｨｩｪｫｬｭｮｯ(0xFF66 to 0xFF6F) Full-width katakana ァアィイゥウェエォオカガキギクグケゲコゴサザシジスズセゼソゾタダチヂッツヅテデトドナニヌネノハバパヒビピフブプヘベペホボポマミムメモャヤュユョヨラリルレロヮワヰヱヲンヴヵヶ (0x30A1 to 0x30F6) Full-width hiragana: ぁあぃいぅうぇえぉおかがきぎくぐけげこごさざしじすずせぜそぞただちぢっつづてでとどなにぬねのはばぱひびぴふぶぷへべぺほぼぽまみむめもゃやゅゆょよらりるれろゎわゐゑをん (0x3041 to 0x3093) Japanese long vowel (cho-on): ー(0x30FC: full-width), ｰ(0xFF70: half-width)
kanji	The specific range of corresponding Kanji characters is decided as below according to the types of the standard characters specified in the “default_ccs” attribute of the book information module <bvf> tag. “JIS X 0201, JIS X 0208:1997”: Dai 1 to Dai 2 Suijun Kanji Sets (Stipulated by JIS X 0208:1997) “JIS X 0201, X-SH-JIS 0213:2004”: Dai 1 to Dai 4 Suijun Kanji Sets (Stipulated by JIS X 0213:2004) The following characters are included regarding the type of standard character sets used. ヽ(0x30FD),ヾ(0x30FE),ゝ(0x309D),ゞ（0x309E）〃（0x3003）,仝（0x4EDD）,々（0x3005）,〆（0x3006）,〇（0x3007）

<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.4.7.9 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".

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

cho_on: Defines the conversion method for the Japanese long vowel (cho-on). Accepts the values listed below. Can be omitted. When omitted, “delete” is set.

"delete" Removes the character.

"repeat" Repeats the vowel of the preceding character.

"no" Does not convert.

daku_on: Defines the conversion method for the Japanese voiced sound (daku-on). Accepts the values listed below. Can be omitted. When omitted, “yes” is set.

"yes" Converts to a voiceless sound, or sei-on.

"no" Does not convert to a voiceless sound, or sei-on.

handaku_on: Defines the conversion method for the Japanese semi-voiced sound (handaku-on). Accepts the values listed below. Can be omitted. When omitted, “yes” is set.

"yes" Converts to a voiceless sound, or sei-on.

"no" Does not convert to a voiceless sound, or sei-on.

soku_on: Defines the conversion method for the Japanese geminate consonant (soku-on or "small tsu") Accepts the values listed below. Can be omitted. When omitted, “yes” is set.

"yes" Converts to an upper case character.

"no" Does not convert to an upper case character.

yo_on: Defines the conversion method for the Japanese palatalized sound (yo-on). Accepts the values listed below. Can be omitted. When omitted, “yes” is set.

"yes" Converts to an upper case character.

"no" Does not convert to an upper case character.

other_small_kana: Defines the conversion method for the Japanese small hiragana and katakana characters other than soku-on and yo-on. Accepts the values listed below. Can be omitted. When omitted, “yes” is set.

"yes" Converts to an upper case character.

"no" Does not convert to an upper case character.

Example:

```
<search_table bookmark=“yes” wordbook=“yes” jump_search_root=“no” jump_search=“yes”
 all_search=“no”>
   <search_table_def id=“ST0001” name=“Japanese dictionary” wild=“yes” blank=“yes”
end=“yes” help_page_id=”PG9999”>
   <enable_key_type number=“no” alphabet=“no” kana=“yes” kanji=“yes”/>
   <key_normalization cho_on=“delete” daku_on=“yes” handaku_on=“yes” soku_on=“yes”
yo_on=“yes”
    other_small_kana=“yes”>
   </search_table_def>
   <search_table_def id=“ST0002”          ・・・>
            :
   </search_table_def>
</search_table>
```

B.4.5 Event info module <event_info>

B.4.5.1 <event_info> tag

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 Defines the playback environment. Only the following values are valid. “single” is used if omitted.

“single” (designated for one-page display)

[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.4.5.2 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 1
 <event>
         <trigger>
                 <trigger_pointer id=" OB0001/CR0002" action_flag="click"/>
         </trigger>
         <action>
                 <action_page_jump page_id="PG0003"/>
         </action>
 </event>                                                     Event2

       …
</event_info>
```

B.4.5.2 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 sound file". This <event> tag has the following child elements.

[Child elements]

<trigger>	See B.4.5.3 for details. There shall be only one <trigger> tag, and it can have only one child element. Cannot be omitted.
<action>	See B.4.5.4 for details. There shall 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>                                                   Trigger
 <action>
         <action_page_jump page_id="PG0003"/>
 </action>                                                    Action

</event_info>
```

B.4.5.3 Trigger information

The <trigger> tag (trigger (pointer) <trigger_pointer>) has the following child elements.

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

[Attributes]

id: points at the target of the trigger. 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.4.7.2.3 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.4.7.2.3) 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.3.15). 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 shall trigger the first of the two events, resulting in a page jump to PG0043.

Example:

```
<trigger_pointer id="OB003k/CR0023" action_flag="click"/>
<trigger_pointer id="OB003k/CR0023"/>
```

B.4.5.4 Action information

B.4.5.4.1 <action> tag

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

B.4.5.4.2 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.4.6) (in the current version of this standard, only possible for sound and movie objects):

Written as an *Object_ID*.

Example:

```
object_id="OB003k"
```

- If the object is inserted in the *text object instance* or in the *dictionary data object instance* by the <object> tag (in this standard, only possible for animations):

Written as *Object_ID*/*Char_ID*.

Example:

```
object_id="OB003k/CR0023"
```

Note that the Char ID part shall be an “id” number registered in the “char_id” attribute of an <object> tag (See B.4.7.2.3).

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

Example:

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

B.4.5.4.3 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 4 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 in the “Telephone_number” format, starting with "tel: " followed by a telephone number.

c) Write a mail to the specified address:

The “book” attribute should be written in the “Mail_address” format, starting with "mailto: " followed by a single email address.

d) Specify a viewer’s special processing:

The “book” attribute should specify a special processing such as placing “jump://” to a string.

book_type: Used to differentiate the types of content stored in the “book” attribute. Must be omitted if book is omitted too, cannot be omitted when book is not omitted.

a) When the “book” attribute specifies a web site.

The attribute shall contain: "text/html"

b) When the “book” attribute specifies a telephone number.

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

c) When the “book” attribute specifies a mail address.

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

d) When the “book” attribute specifies a special processing to the viewer.

The attribute shall contain: "application/x-e-book-viewer"

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 a) website, b) telephone, c) mail address, or d) viewer’s special processing, 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 or dictionary data object of flow data (see B.4.4.2.4). In that case, the position within that text in the *text object* or *dictionary data object instances* 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 a) website, b) telephone, c) mail address, or d) viewer’s special processing, 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.4.6 Parts data module <parts_module>

B.4.6.1 Storage and management

This section 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.

<dict_data_object_entry> Records dictionary data objects.

Example:

```
<parts_module>
        <object_table>
                <dynamic_text_object_entry object_id="OBT001" ... >
                        ...
                </dynamic_text_object_entry>
                <dynamic_text_object_entry object_id="OBT002" ... >
                        ...
                </dynamic_text_object_entry>
                <sound_object_entry object_id="OBS001" ...>
                                        …

        </sound_object_entry>
                <dynamic_text_object_entry object_id="OBT003" ... >
                        ...
                </dynamic_text_object_entry>
                ...
                <search_page_object_entry object_id="OBSP01" ... >
                        ...
                </search_page_object_entry>
                ...
                <movie_object_entry object_id=”OBMV01” … />
                …
        </object_table>
</parts_module>
```

Subclauses B.4.6.2 to B.4.6.7 below will explain how each object is registered.

B.4.6.2 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 shall be checked against the “type” attribute.

type: Stores the MIME type of the *text object instance*. The only possible value is "text/x-bvf-text". Cannot be omitted.

layout_info_src: *Filename* of the layout pattern data list (see B.4.7.4). Written in the standard *Filename* format Can be set only when the object specified by the “object_id” is a “*text object instance 2*” (see B.4.7.3). Can be omitted. When omitted, the display follows the old format.

layout_info_type: Stores the MIME type of the layout pattern data list. The only possible option is "text/x-bvf-layout". Cannot be omitted if the “layout_info_src” attribute is specified.

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.4.6.3 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 this standard, the following types are permitted. Cannot be omitted.

MP3 format:	“audio/mp3”
AAC format:	“audio/3gpp2”
SMAF format:	“application/x-smaf”
3GPP format:	“audio/3gpp”
MP4 format:	“audio/mp4”

src: *Filename* of the sound object, written in the standard *Filename* format. Cannot be omitted. Before opening the file specified in this attribute, it shall be checked against the “type” attribute. In this standard, the compatible formats are limited to MP3 (extension: .mp3), AAC (extension: .3g2), SMAF (extension: .mmf), 3GPP (extension: .3gp), and MP4 (extension: .mp4).

[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.4.6.4 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 *Filename* format. Cannot be omitted. Before opening the file specified in this attribute, it shall be checked against the “type” attribute.

type: Records the MIME type of the search page object. In this standard, 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 flow data or the event data 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:

```
<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.4.6.5 Movie object <movie_object_entry>

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

[Attributes]

src: *Filename* of a movie object, written in the standard *Filename* format. Cannot be omitted. Before opening the file specified in this attribute, it shall be checked against the “type” attribute. In this standard, the compatible formats are limited to 3GPP2 (extension: .3g2), 3GPP (extension: .3gp), and MP4 (extension: .mp4).

type: Records the MIME type of the movie object. In this standard, the following format is permitted. Cannot be omitted.

3GPP2 format:	“video/3gpp2”
3GPP format:	“video/3gpp”
MP4 format:	“video/mp4”

object_id: Assigns an ID number to the object. This number is used by the event data, etc. 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.4.6.6 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 shall 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.4.6.7 Dictionary data object <dict_data_object_entry>

The area for recording the objects in the dictionary data format. *Dictionary data object instance* registration module <dict_data_object_entry> has the following attributes and child elements.

[Attributes]

src *Filename* for the *dictionary data object instance*. Written in the standard *Filename* format. Cannot be omitted. Before opening the file specified in this attribute, it shall be checked against the “type” attribute.

type Stores the MIME type of the *dictionary data object instance*. The only possible option is “text/x-bvf-dict-data”. Cannot be omitted.

object_id The object_id number to be set for this object. This number is used from the flow data or event data modules to refer to this object. Written in the *Object_ID* format. Cannot be omitted.

[Child elements]

<permission_info> Defines the permissions for the *object instance* specified by the "src" attribute. Written in the standard *Permission* format. Can be omitted.

Example:

```
<parts_module>
   <object_table>
      <dict_data_object_entry src=”dict1.xml” type=”text/x-bvf-dict-data” object_id=”OBD001”>
      <permission_info> ・・・＜／ｐｅｒｍｉｓｓｉｏｎ＿ｉｎｆｏ＞
      </dict_data_object_entry>
         :
   </object_table>
</parts_module>
```

B.4.7 Object instances

B.4.7.1 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 types of *object instances* are listed below:

Text object instance

Text object instance 2

Image object instance

Sound object instance

Animation object instance

Search page object instance

Movie object instance

Comic object instance

Dictionary data object instance

B.4.7.2 Text object instance <text_data>

B.4.7.2.1 General

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 the other for storing the text itself (<text_body>).

If a passage “expressed in ‘em’” appears in this document, a relative value to “1” (default size) is used. For example, “0.5em” indicates the half size of the default font.

B.4.7.2.2 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.4.4.2.3, 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.4.4.2.4). 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.

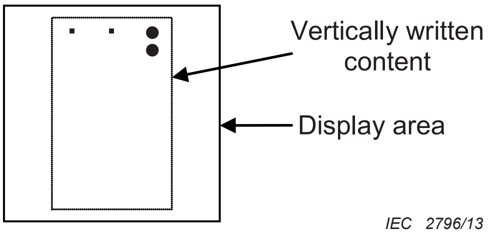


Figure B.2 – Example of valign=”middle”

[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 *color* data format. When omitted, defaults to the <flow_default_attribute> value.

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

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

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

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

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

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

[Attribute]

color_space, color, opacity:

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

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

"image/png"

"image/jpeg"

“image/x-ccf”

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

[Child elements]

<permission_info>:

Defines the permissions about the image referred to by the src attribute. Written in the standard *Permission* format. 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”

"audio/3gpp"

"audio/mp4"

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 shall be checked against the type attribute.

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

"yes"	Played back in a loop.
"no"	Is not played back in a loop.

[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>/(alternatively) <kinsoku> 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, it defaults to the <flow_default_attribute> value. This tag has the following attributes and child elements.

[Attributes]

method: Choose the line breaking method. It defaults to "none". Possible values are:

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

"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.

hanging_punctuation: Activates or deactivates the processing of hanging characters. Possible values are "yes" and "no". It 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.

[Child elements]

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

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

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

Example:

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

B.4.7.2.3 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 start the new line after the image, <br clear="all"/> shall be explicitly used. <p> has the following attributes and child elements.

[Attributes]

top_line_indent: Sets the size (expressed in “em”) of the indentation of the first line of the paragraph. The spaces are inserted before the first character following the <p> tag, and the first character following a <br/> tag in the paragraph. The unit shall 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. When omitted, behavior depends on the viewer. Note that, this attribute can be set to a negative value, like "-2em". In this case, the top of the paragraph begins at the position moved "2em" beyond the line top position specified by the "top" attribute.

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 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.

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.

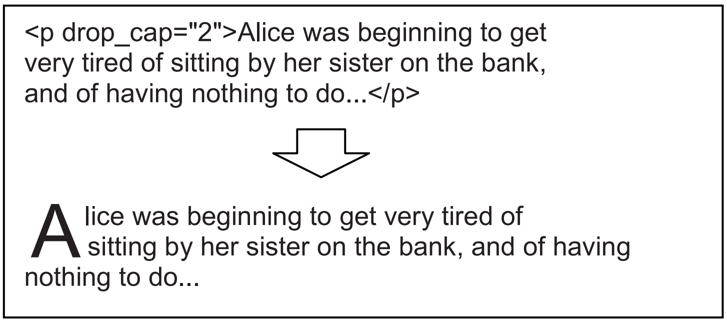
align: Determines whether the text should be in the center, left-aligned or right-aligned within the line size defined by the top, bottom and first line indent attributes. This attribute accepts the values listed below. When omitted, the current setting is kept unchanged.

"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.



<p drop_cap="2">Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do...</p>

Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do...

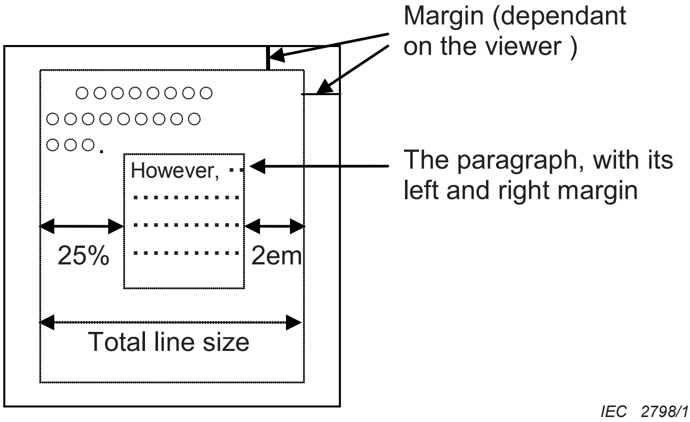
IEC 2797/13

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".


Margin (dependant on the viewer)
○○○○○○○○○
○○○○○○○○○○
○○○.
However,
The paragraph, with its left and right margin
25%
2em
Total line size

IEC 2798/13

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>/(alternatively) <telop> Defines a line of scrolling text. Line breaks are automatically inserted around this element (right before the <scrolling_text>/<telop> tag and right after the <scrolling_text>/<telop> 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 scrolling 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>/ <telop> 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>, <font>, <horizontal>/<yoko>, <ruby> and <object>. However, the <object> tag has the following additional limitations when used as a child element of the <scrolling_text>/<telop> 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 cannot be used.

Example:

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

b) Inline elements

<br/> 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 <br/> tag does not do so when this attribute is omitted. The following values are possible.

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

Example:

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

<hr/> Draws a horizontal line (a vertical one when the text is written vertically). As this line shall always be alone in its row (column when the text is vertical), a line break (equivalent to <br/>) 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". It defaults to center when omitted.

Example:

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

<font>...</font> 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 <font> 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 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, defaults to "last".

"last": Same size as the character before the <font> 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 <font> tag is used.

bold: Defines whether the string included in the <font> 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 <font> tag should be underlined or not. However, objects included inside the <font> 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 <key_entry> and <page_break/>) and <object> can be used.

Example:

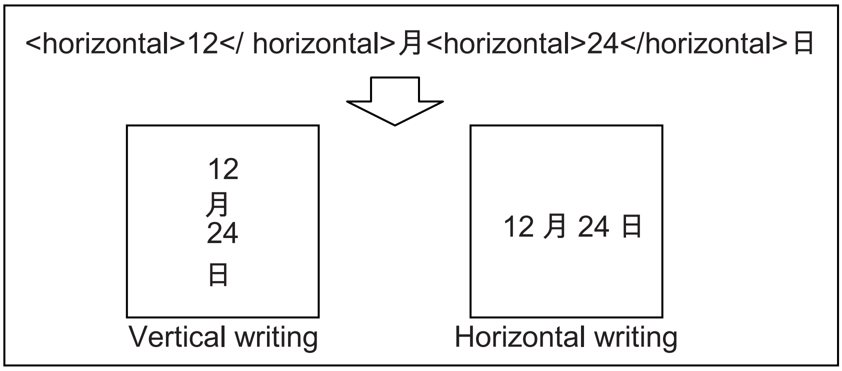
```
<font name="Aaa sans serif">.....</font>
<font size="200%">......</font>
```

<horizontal>...</horizontal>/(alternatively) <yoko>...<yoko> 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:



```
<horizontal>12</ horizontal>月<horizontal>24</horizontal>日
```

IEC 2799/13

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.

[Attribute]

None

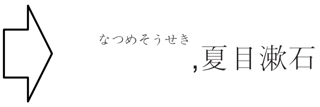
[Child elements]

<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:

```
<ruby>
        <rbase>夏目漱石</rbase>
        <rtop>なつめそうせき</rtop>
</ruby>
```



IEC 2800/13

Figure B.6 – Ruby

<external_char> Inserts an *External character*. The detailed syntax is described in B.3.12. 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="鴎"/>
```

<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 shall 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 apperance 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 <br>, <font>, <horizontal>, <ruby> and <object> tags. However, the following restrictions apply to the <object> tag:

The src attribute shall point to an image file.

The char_id attribute 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 shall be at least one <key_item> per <key_entry>. It has the following attributes and child elements.

[Attributes]

table_id: Specifies the search table to which this keyword is to be added, by using the table's ID number, which is defined in <seach_table_def> (see B.4.4.2.6). 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.4.4.2.6. 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 ...
```

<meaning_start/> Specifies what should be displayed when “browsing” key is pressed. When this happens, viewer shows the text after this tag either up to the next tag that represents a new line or it meets the end of the line. Ignored if the browsing key is not pressed or the viewer has no such key.

Example: (XML)

```
<meaning_start/> meaning 1)  [Noun] ... Examples of
common expression using this word are as follows.
   <br/>
<p top="1em">
Common expression  1) ... <br/>
Common expression  2) ... <br/>
</p>
<meaning_start/>  meaning  2)   [Counter  suffix]  ...
<br/>
```

When the browsing key is pressed: (display)

```
Meaning  1)  [Noun]  ...  Examples  of  common
expression      using this word are as follows.
Meaning 2) [Counter suffix]
```

Otherwise: (display)

```
Meaning  1)    [Noun]  ...  Examples  of  common
expression      using this word are as follows.
  Common expression  1) ...
  Common expression  2) ...
Meaning 2) [Counter suffix] ...
```

<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.4.4.2.4).

head_button_control: Permits or forbids moving from a flow to the next/previous flow by pressing a headword button prepared for some viewer. The following options are available.

"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.

When this attribute is omitted, the behavior should conform to the overall setting defined for the *text object instance* (for the “head_button_control” attribute of the <flow_data> tag, see B.4.4.2.4).

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:

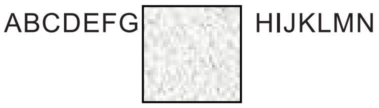
Png image:	"image/png"
Jpeg image:	"image/jpeg"
CCF image:	“image/x-ccf”
Animation object:	"application/x-bvf-flip-animation" (see B.4.7.7)

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

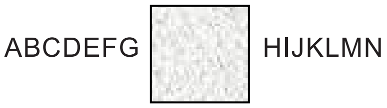
char_id: Assigns a string ID number to this object, written as a *Char_ID*. Can be omitted.

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.

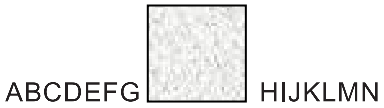
"top": Inline display. When text is written horizontally, the topmost part of the row should be aligned with the upper edge of the object; when written vertically, the rightmost part of the column should be aligned with the right edge of the object.



"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.



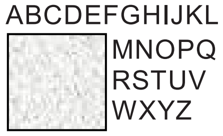
"bottom": Inline display. When text is written horizontally, the base line of the row should be aligned with the lower edge of the object; when written vertically, the base line of the column should be aligned with the left edge of the object.



"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.

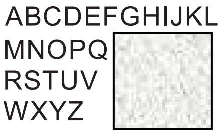
The following example should be rendered the same way with the <object> tag at any position between after the letter "A" and before the letter "N" in the XML file.



"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 "N" in the XML file.



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.4.5.4.2) 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.4.5.2).

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.4.7.3 Text object instance 2 <text_data2>

B.4.7.3.1 General

If the layout for <text_data2> is specified by using <layout_info>, <text_default_attribute> setting is not allowed, and instead, information corresponding to <text_default_attribute> is stored. If the layout is not specified, <text_default_attribute> setting is allowed in <text_data2> just as allowed in <text_data>.

B.4.7.3.2 Text body 2 <text_body2>

B.4.7.3.2.1 General

The area for storing the content of the text object instance 2. Tags that can be written in the <text_body2> folder are the same as written in the <text_body>. Note that features are added to the exising tags as explained below. Turn to <text_body> for existing features for those tags.

B.4.7.3.2.2 Paragraph

<p>

[Attributes]

align Below are the attribute options added for <text_body2>.

"justified": Displays the character strings and objects in the equal spacing layout.

"justified_center": Displays the character strings and objects in the equal spacing layout, in the center.

"justified_right": Displays the character strings and objects in the equal spacing layout, right-aligned when written horizontally and downward-aligned when written vertically.

"justified_left": Displays the character strings and objects in the equal spacing layout, left-aligned when written horizontally and upward-aligned when written vertically.

B.4.7.3.2.3 Inline elements

<br/> Below are the attribute options added for <text_body2>.

[Attributes]

width: Specifies the line width, by a numerical value using the unit of "em". It is effective when there exists only the <br/> tag with the line width specified to this value within one line.

<hr/> New specifications are added to the “size”, “length”, and “align” attributes by their feature expansion. New attributes such as “h_src” “v_src” and “type” and new child elements are also added.

[Attributes]

size: Specifies the thickness of a line, by a numerical value using the unit of “em”. If omitted, the viewer acts in the same way as “0.1em” is specified. This attribute setting is ignored if an image has been defined by the “h_src” or “v_src” attributes.

length: Specifies the length of a line, by either a numerical value using the unit of “em” or the percentage value toward the line-width/line-length (in horizontal/vertical writing) of a line in which this <hr/> tag exists. If omitted, the viewer acts in the same way as “100%” is specified. The size of a line (width/height) does not necessarily match the size of the displayed area. The size of a line (width/height) may be narrower than the displayed area width because of the previous tag settings (such as top/bottom attributes of <p>) stored before this <hr/> setting. The image will be displayed repeatedly in order for the area to fit in this specified area, even if the image is specified by the “h_src” or “v_src” attributes.

align: Specifies the location of a line. It is effective when the line length is shorter than the line width. Possible values are “left”, “center” and “right”. If omitted, the viewer acts in the same way as “center” is specified. The image will be displayed as specified by the “align” attribute, even if the image is specified by the “h_src” or “v_src” attributes.

v_src: Sets a portrait type fancy frame image. A pattern of an image is specified in the standard *Filename* format. This image setting will disable the “size” attribute settings. The viewer will repeatedly display the image. The viewer will display the specified file in the same manner as in each portrait/landscape display mode without rotating the file. If no switching setting between portrait and landscape displays is required, for example in ○, the setting for either of the “v_src” or “h_src” will be enough. If the “baseline” attribute of the <layout_info> tag is set, only an image file specified by one appropriate attribute can be used, even if both “v_src” and “h_src” are specified. The possible image file formats are PNG format and JPEG format in this standard. Cannot be omitted if the “type” attribute is specified. If both “h_src” and “type” attributes are omitted, nothing is specified. Before opening the file specified in this attribute, it shall be checked against the “type” attribute.

h_src: Sets a landscape type fancy frame image. A pattern of an image is specified in the standard *Filename* format. This image setting will disable the “size” attribute settings. The viewer will repeatedly display the image. The viewer will display the specified file in the same manner as in each portrait/landscape display mode without rotating the file. If no switching setting between portrait and landscape displays is required, for example in ○, the setting for either of the “v_src” or “h_src” will be enough. If the “baseline” attribute of the <layout_info> tag is set, only an image file specified by one appropriate attribute can be used, even if both “v_src” and “h_src” is specified. The possible image file formats are PNG format and JPEG format in this standard. Cannot be omitted if the “type” attribute is specified. Nothing is set for both “v_src” and “type” attributes if they are omitted. Before opening the file specified in this attribute, it shall be checked against the “type” attribute.

type: Stores the MIME type of the ruled image. The following formats are available in this standard. Cannot be omitted if the “src” attribute is specified.

PNG format:	"image/png"
JPEG format:	"image/jpeg"

[Child elements]
<permission_info> Defines the permissions for the fancy frame imagespecified by the "src" attribute. Written in the standard *Permission* format. Can be omitted. If no fancy frame image is set for the “src” attribute, this permission information should be ignored.

<select> Switches the character strings to be displayed according to the settings such as the baseline direction of the text. This tag has the following attributes and child elements.

[Attributes]
variable Sets the parameter for switching the display. The only possible option is “baseline” in this standard. Cannot be omitted.

[Child elements]
<select_item> Defines the character strings to be displayed. This tag has the following attributes and child elements.

[Attributes]
type Sets the parameter pattern specified by the “variable” attribute. When the viewer setting selects this value, the content of the child elements is displayed.

"right"	Horizontal writing
"down"	Vertical writing

[Child elements]
For child elements, following tags can be set: extended character strings, <external_char>, <font>, <ruby>, <horizontal>/<yoko>, and <object>.

B.4.7.3.2.4 Object

<object> The “position” attribute is newly added.

[Attributes]
position Defines the display position of the object. The only possible option is "on_edge". This tag is displayed at the beginning/end of the paragraph. Can be omitted. Displayed at the position where the <object> tag is inserted as in the conventional manner when omitted. Setting the “align” attribute and the “position” at the same time is not allowed.

“on_edge” Displays the object on the edge of a paragraph (beginning/end) below the position the <object> tag is inserted. If the original size of the object does not fit into one paragraph, it is resized so that it fits into one paragraph. If it does fit into one paragraph, the object is displayed at the center of the screen without being resized. Note that if a content with no column setting (i.e., no

<layout_default_columns> tag) is displayed on the viewer with the line spacing control feature, this setting is ignored and processed as if the “align” attribute is omitted (“bottom” is applied on the horizontal writing display and “middle” is applied on the vertical writing display).

[Child elements]
<permission_info> Defines the permissions for the *object instance* specified by the "src" attribute. Written in the standard *Permission* format. Can be omitted.

B.4.7.4 Layout pattern data list <layout_info_list>

B.4.7.4.1 <layout_info_list> tag

The area for storing the layout pattern and the display setting of each area when displaying this *text object instance 2*. Specifies this layout pattern list for each *text object instance 2* (see B.4.6.2). In order for the layout pattern data list to change the layout patterns and display settings appropriately according to the device specifications or current display settings, multiple layout pattern data with the information below can be written. The viewer is able to select information to use for the display according to the condition when displaying.

Layout pattern data can include the following information.

Device information

Outline information

Dynamic text information

Object information

In addition, as was explained in B.4.4.2.3 some of the parameters defined here may also be set in the whole text flow (defined by the “body_id” attribute of the <flow_data> tag, and values are set in the attributes of the <flow_default_attribute> tag. See B.4.4.2.3 and B.4.4.2.4). These are used as default if the local ones (the ones defined here) are not set when the “body_id” attribute of the <flow_data> tag is defined as the “*text object instance 2*”. In case both are defined, the local settings take precedence.

The layout pattern data list <layout_info_list> has the following child elements.

[Child elements]
<layout_info> The area for storing one layout pattern. See B.4.7.4.2 for details about the format of this tag.

B.4.7.4.2 Layout pattern data <layout_info>

The area for storing layout patterns. Can specify plurally. Follow the limitations below when specifying plurally.

- Cannot write plural <layout_info> tags with the same attribute at the same time.
- Cannot write a <layout_info> tag with the “baseline” attribute specified and a <layout_info> tag without the “baseline” attribute specified at the same time.

This tag has the following attributes and child elements.

[Attributes]
display_size Specifies the display size, written in the standard coordinates format. Cannot be omitted.

baseline Sets the baseline direction. The following values can be used: Can be omitted. The user should be able to select between vertical written text display and horizontal written text display, when omitted.

"right"	Horizontal writing
"down"	Vertical writing

[Child elements]

<outline_info> The area for storing the outline information. See B.4.7.4.3 for details about the format of this tag.

<layout_default_attribute> The area for storing a default attribute of the text to be used in the area specified by the <dynamic_text_area> tag. See B.4.7.4.6 for details about the format of this tag.

<object_area> The area for storing the object information. See B.4.7.4.5 for details about the format of this tag.

<dynamic_text_area> The area for storing the dynamic text information. See B.4.7.4.4 for details about the format of this tag.

Note that if each area is specified by the <object_area> and the <dynamic_text_area> tags, the following limitations are applied. If exceeds the limit, converter error occurs.

A) Defines each area as it fits inside the rectangle specified by the “rect” attribute of <outline_info>. Placing the area frame on the line of the rectangle is allowed.

B) Defines each area as it does not overlap other areas. Area frames cannot be overlapped.

C) The following limitations for the size (height/width) are applied if plural <dynamic_text_area> tags are set. This is for the column settings display.

If the “baseline” attribute of <layout_info>, its parent attribute, is “down”

– Align the height of all the areas.

If the “baseline” attribute of <layout_info>, its parent attribute, is “right”

– Align the width of all the areas.

If the “baseline” attribute of <layout_info>, its parent attribute, is omitted.

– Align the height and the width of all the areas. The height and the width do not necessarily have to be the same.

D) Setting the plural <dynamic_text_area> tags and the <layout_default_columns> tag at the same time is not allowed.

B.4.7.4.3 Outline information <outline_info>

B.4.7.4.3.1 <outline_info> tag

The area for storing the overall information of the area that is displayed by this layout pattern, among all the displays of the device. Can be omitted. This tag has the following attribute and child elements.

[Attributes]

rect Defines the circumscribed rectangle’s coordinate that contains all the areas included in the <layout_info> displayed, in the “Region” format. Can be omitted. When omitted, the coordinate value whose size is the same as the display size that is specified by the “display_size” attribute of the <layout_info> tag is stored in the content at the time of conversion. The coordinate value set here is the value on the coordinate system based on the displayed screen size.

[Child elements]

<outline_polygon> The area for storing the apex coordinate of the outline. See B.4.7.4.3.2 for details about the format of this tag.

<outline_background> The area for storing the background image of the outline. See B.4.7.4.3.3 for details about the format of this tag.

<outline_background_music> The area for storing the background music of the outline. See B.4.7.4.3.4 for details about the format of this tag.

B.4.7.4.3.2 The apex coordinate of the outline <outline_polygon>

Defines the outline’s color and thickness, and also the apexes of the polygon. This tag has the following attributes and a child element. If both the “line_color” and “line_width” attributes are omitted, the line of the outline should not be displayed.

[Attributes]
line_color Specifies the color of the outline’s line. Can be omitted. If omitted, the behavior depends on the viewer.

line_width Specifies the thickness of the outline’s line. Expressed in numerical value with points or dots. Can be omitted. If omitted, the behavior depends on the viewer.

[Child elements]
<vertex> Defines the apexes of a polygon which composes the outline. Can be omitted. When omitted, the coordinate value (the same as the circumscribed rectangle’s value specified in “rect” attribute of the <outline_info> tag) is recorded during the conversion. Diagonal polygon is not accepted. The coordinate value should be written in the standard coordinates format according to the display (screen) size. Note that only rectangle is supported in the current specification.
This tag has the following attributes.

[Attributes]
position Specifies the apexes of a polygon that composes the outline. Written in the standard Coordinates format.

B.4.7.4.3.3 Outline background image <outline_background>

Sets the outline background image and its color. This tag has the following attributes and child elements.

[Attributes]
color_space, color, opacity Defines the default background color of the outline, in the standard *Color* format. This setting cannot be changed by the user. Can be omitted. Not specified if the “Color” attribute is omitted.

type Stores the MIME type of the background image. The following formats are available in this standard. Cannot be omitted if the “src” attribute is specified.

PNG format:	"image/png"
JPEG format:	"image/jpeg"
CCF format:	"image/x-ccf"

src *Filename* for the background image. Written in the standard *Filename* format. The choices are limited to the following format image files in this standard: PNG, JPEG, and CCF. Cannot be omitted if the “type” attribute is specified. Nothing is set for both “src” and “type” attributes if they are omitted. Before opening the file specified in this attribute, it shall be checked against the “type” attribute.

[Child elements]
<permission_info> Defines the permissions for the background image specified by the "src" attribute. Written in the standard *Permission* format. Can be

omitted. If the "src" attribute does not specify the background image, this permission information should be ignored.

B.4.7.4.3.4 Outline background music <outline_background_music>

Defines the background music tag to be played back when displaying this outline. If this tag is omitted, no background music should be played. This tag has the following attributes and child elements.

[Attributes]

type Stores the MIME type of the background music. The following formats are available in this standard. Cannot be omitted.

MP3 format:	"audio/mp3"
AAC format:	"audio/3gpp2"
SMAF format:	"application/x-smaf"
3GPP format:	"audio/3gpp"
MP4 format:	"audio/mp4"

src *Filename* of the background music. Written in the standard *Filename* format. The settable files are limited to the following image formats in this standard: MP3 (with ".mp3" extension), AAC (with ".3g2" extension), SMAF (with ".mmf" extension), 3GPP (with ".3gp" extension), and MP4 (with ".mp4" extension). Cannot be omitted. Before opening the file specified in this attribute, it shall be checked against the “type” attribute.

loop Defines whether the background music specified in the “src” attribute is played back repeatedly, selecting any of the values below. Can be omitted. “no” is set if omitted.

"yes"	Played back in a loop.
"no"	Is not played back in a loop.

[Child elements]

<permission_info> Defines the permissions for the background music specified by the "src" attribute. Written in the standard *Permission* format. Can be omitted.

B.4.7.4.4 Dynamic text information <dynamic_text_area>

The area for storing the area information to display the text entity data 2. Can be omitted. When this tag is omitted, text entity 2 with no content shall be specified by the <dynamic_text_object_entry> tag (see B.4.6.2). More than one elements can be applied. In that case, the elements should be read in the recorded order.

This tag has the following attributes.

[Attributes]

rect Defines the area where *text object instance* is used, in the “Region” format. The coordinate value should be defined according to the “display_size” attribute of the <layout_info> tag. Cannot be omitted.
The coordinate value should be defined according to the display (screen) size.

bg_color, bg_color_space, bg_opacity
Defines the default background color, in the standard *Color* format. Can be omitted. If omitted, the background color information of the <outline_background> tag is used.

B.4.7.4.5 Object information <object_area>

Defines the constantly-displayed object such as an image, and its display area, in the “Region” format. Can specify the image format file. Displays the object as is (without changing the size) in the center of the area. When the object's size exceeds the area, only the part of the object that falls within the area should be displayed. Can be omitted. Can specify plurally. This tag has the following attributes and child elements.

[Attributes]

rect Defines the area of constantly-displayed objects, in the "Region" format. The coordinate value should be defined according to the "display_size" attribute of the <layout_info> tag. Cannot be omitted.

src The object filename to be displayed in the area. Written in the standard *Filename* format. Cannot be omitted. The settable file is an image in PNG format, JPEG format, and CCF format.

type Stores the MIME type of the object file. The following formats are available in this standard. Cannot be omitted.

PNG format:	"image/png"
JPEG format:	"image/jpeg"
CCF format:	"image/x-ccf"

bg_color, bg_color_space, bg_opacity
Defines the default background color, in the standard *Color* format. If the "color" attribute is omitted, the background color information of the <outline_background> tag is used.

char_id Assigns a string ID number to this object area. Written in the standard *Char_ID* format. Can be omitted if this tag is not set. When setting this string ID number to an event, take note to the followings: (1) the event cannot be set as a destination page in page jump; (2) the event cannot be set as the trigger/action toward the dynamic text area.

[Child elements]

<permission_info>
Defines the permission for the files specified by the "src" and "alt_src" attributes. Written in the standard *Permission* format. Can be omitted.

B.4.7.4.6 Default settings of the dynamic text information <layout_default_attribute>

B.4.7.4.6.1 <layout_default_attribute> tag

The area for various settings regarding the text entity data 2 to be used in the area specified by the <dynamic_text_area> tag. Can be omitted. This tag has the following attributes and child elements.

[Attributes]

valign Determines how the text box is to be positioned. Accepted only when there is one <dynamic_text_area> tag and the displayed content fits inside the display area, and otherwise this is ignored. The following values are allowed. Can be omitted. When omitted, the text should be displayed from the top of the display area.

"middle"	Displayed in the center within the display area.
"end"	Displayed at the end within the display area.

[Child elements]

<layout_line_breaking_method>
The area for storing the line-breaking specification. See B.4.7.4.6.2 for details about the format of this tag.

<layout_default_size>
The area for storing the letter space, line space, and margins. See B.4.7.4.6.3 for details about the format of this tag.

<layout_default_paragraph>

The area for storing the paragraph specification. See B.4.7.4.6.4 for details about the format of this tag.

<layout_default_font>
The area for storing the font specification. See B.4.7.4.6.5 for details about the format of this tag.

<layout_default_columns>
The area for storing the column specification. See B.4.7.4.6.6 for details about the format of this tag.

B.4.7.4.6.2 Line-breaking specifications <layout_line_breaking_method>

Defines the recommended line-breaking specifications for display the text entity data 2. Can be omitted. When omitted, it defaults to the <flow_default_attribute> value. This tag has the following attributes and child elements.

[Attributes]

method Defines the method for the line-breaking processing. Can be omitted. Not specified if this tag is omitted.
"none" No line-breaking applied.
"run_down" Line-breaking applied (characters that shall not appear at the beginning/end of a line are driven to the previous/next line)

hanging_punctuation Defines whether processing of hanging characters is activated when a target character exists at the end of the line. Can be omitted. Not specified if this tag is omitted.
"yes" Processing of hanging characters is activated.
"no" Processing of hanging characters is disabled.

[Child elements]

<top_prohibit_char> Stores the characters that shall not appear at the beginning of a line. Listed as an extended character string in the "、。.," format. It defaults to an empty list when omitted.

<end_prohibit_char> Stores the characters that shall not appear at the end of a line. Listed as an extended character string in the “、。.,” format. Not specified if this tag is omitted.

<hanging_char> Stores the character strings to be displayed as hanging punctuation. Listed as an extended character string in the “、。.,” format. Not specified if this tag is omitted.

B.4.7.4.6.3 Setting letter spacing, line spacing, and margin size <layout_default_size>

Defines the letter spacing, line spacing, and margin size when displaying the text entity data 2. This tag has the following attributes. Can be omitted. If any of the following attribute is omitted, the behavior depends on the viewer.

[Attributes]

letter_spacing Letter spacing is specified by the percent value based on the font size displayed on the viewer screen. Can be omitted. If omitted, the behavior depends on the viewer.

line_pitch Line pitch is specified by the percent value based on the font size displayed on the viewer screen. Can be omitted. If omitted, the behavior depends on the viewer.

margin Specifies the margin size (top/bottom/left/right) in the display area. The only possible option is a dot value. Can be omitted. If omitted, the behavior depends on the viewer.

B.4.7.4.6.4 Paragraph setting <layout_default_paragraph>

Defines the paragraph settings to display text entity 2 data. Each flow is regarded as one large paragraph. Can be omitted. When omitted, it defaults to the following attributes.

This tag has the following attributes.

[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 <br/> tag in the paragraph. The unit shall 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, behavior depends on the viewer.

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, it 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.

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, it 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. 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 in the case of horizontal writing, or bottom-aligned in the case of vertical writing.

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

"justified" Character strings or objects are displayed in equal spacing layout.

"justified_center" Character strings or objects are displayed in equal spacing layout, in center-aligned.

"justified_right" Character strings or objects are displayed in equal spacing layout, in right-aligned if in horizontal writing (bottom-aligned if in vertical writing).

"justified_left" Character strings or objects are displayed in equal spacing layout, in left-aligned if in horizontal writing (top-aligned if in vertical writing).

B.4.7.4.6.5 Font setting <layout_default_font>

Defines the recommended font name, font size, etc., when displaying the text entity data 2. Can be omitted. This tag has the following attributes.

[Attributes]

fontname Default font name. More than one fonts can be specified. In this 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 settings for each content data are omitted, the behavior depends on the viewer.

fname_change_flag Defines whether the user is allowed to change settings. Only "yes" (font name can be changed by the user) is allowed in this standard. Can be omitted. "yes" is set when omitted.

fontsize Defines the default font size. Expressed in numerical value with points or dots. Can be omitted. If omitted, the behavior depends on the viewer.

bold_flag Defines whether the content data should be displayed as bold or not. The following options are allowed. If omitted, the behavior depends on the viewer. If set to "yes", all the characters shall be displayed in bold style, except the characters that are specified with its "bold" attribute set to "no" by the <font> tag within the *text object instance 2*.

"yes"	Displays as bold.
"no"	Displays normally.

color_space, color, opacity

Defines the recommended character color when displaying the body data. This setting cannot be changed by the user. Defined in the standard *Color* format. If the "color" attribute is omitted, the behavior depends on the viewer.

rubysize Ruby size is specified by the percent value based on the font size displayed on the viewer screen. Can be omitted. If omitted, the behavior depends on the viewer.

ruby_flag Defines whether ruby in the content data is to be viewed or not. The following values can be used: If omitted, the behavior depends on the viewer. If set to "yes" or "yes_only", ruby is displayed for the character strings sandwiched by the <ruby> tags in the *text object instance*.

"yes"	Ruby should be displayed, but can be turned off by the user.
"yes_only"	Ruby shall be displayed. This option cannot be turned off by the user. However, this does not apply to viewers that are not able to display ruby.
"no"	It is recommended that ruby not be displayed, but can still be turned on according to the user preferences.
"no_only"	Ruby shall not be displayed, and cannot be turned off by the user. However, this does not apply if the viewer is incapable of disabling ruby display.

fsize_autochange_flag When this flag is on, font size should be changed automatically in the same ratio between the screen size of the viewer and the screen size defined by the "display_size" attribute of the <layout_info> tag. When off, displayed as is. It defaults to "off", if omitted.

"on" Changes font size automatically in the same ratio between the screen size of the viewer and the screen size defined by the “display_size” attribute of the <layout_info> tag.

"off" Displays as is.

B.4.7.4.6.6 Column specification <layout_default_columns>

When this tag is applied, the viewer automatically decides the number of columns to display according to each value of the following attributes. If this tag is omitted, column should not be displayed. This tag has the following attributes and child elements.

[Attributes]

line_character_max_count

Defines the maximum number of characters in one line. Can be omitted. If omitted, the behavior depends on the viewer.

line_character_min_count

Defines the minimum number of characters in one line. Can be omitted. If omitted, the behavior depends on the viewer.

tray_spacing

Line spacing is specified by the percent value based on the font size displayed on the viewer screen. Can be omitted. If omitted, the behavior depends on the viewer.

[Child elements]

<columns_ruled_line>

Defines the ruled line settings between columns. If this tag is omitted, there should not be ruled lines. This tag has the following attributes and child elements.

[Attributes]

width Determines the ruled line’s width. Expressed in numerical value with points or dots. Can be omitted. If omitted, the behavior depends on the viewer. This setting is disabled if a ruled image is set by the “src” attribute.

color_space, color, opacity

Determines the ruled line’s color. Defined in the standard *Color* format. Can be omitted. If omitted, the behavior depends on the viewer. This setting is disabled if a ruled image is set by the “src” attribute.

align Sets the position of the ruled line. It is effective when line width is shorter than the line length. Accepts the values listed below. Can be omitted. If omitted, the viewer acts in the same way as “center” is specified.

"top" Displayed left-aligned for horizontal writing, top-aligned for vertical.

"center" Displayed in the center.

"bottom" Displayed right-aligned for horizontal writing, bottom-aligned for vertical.

length Determines the ruled line’s length. Expressed in "em" or as a percentage of the column’s width (for horizontal writing, the height). If omitted, the viewer acts in the same way as “100%” is specified.

v_src Defines an image of the ruled line for vertical writing. A pattern of an image is specified in the standard Filename format. This attribute takes precedence over both the “width” and “color_space, color, opacity” attributes. The viewer will repeatedly display the image. In this case, the viewer will display the defined file as is (without rotating) according

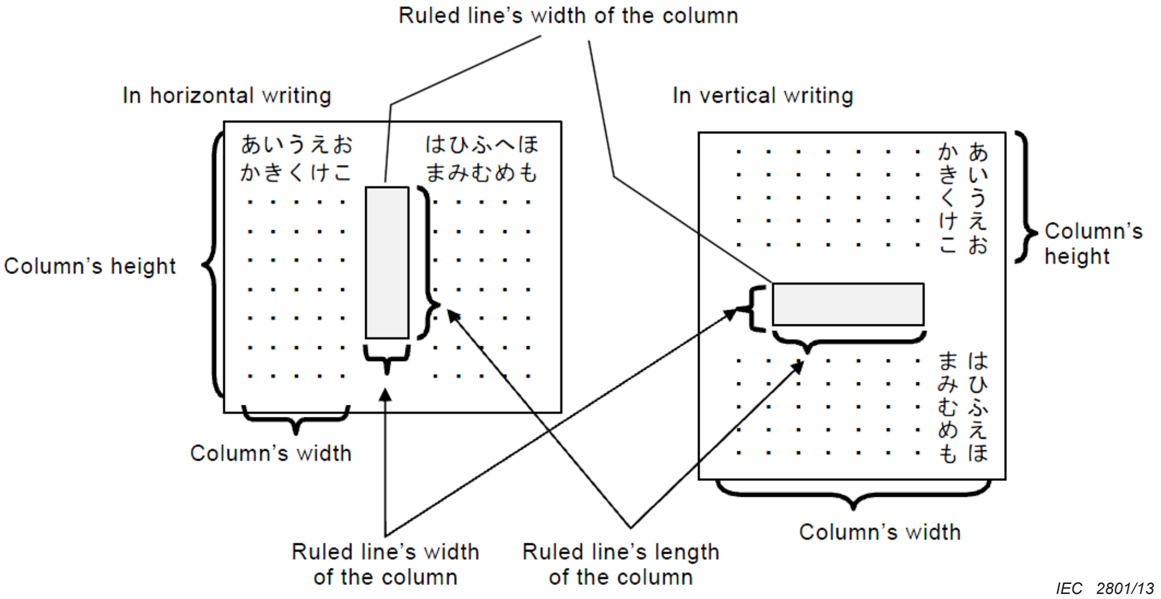
to the writing direction even if the drawing area is cut off in the middle of a pattern. When the viewer recognizes the direction of writing automatically, only either the “v_src” or “h_src” attribute has to be defined. Note that the ruled line itself will be displayed horizontally (for vertical writing).
The PNG format and JPEG format image files are selectable in this standard. Cannot be omitted if the “type” attribute is specified. Nothing is set for both “src” and “type” attributes if they are omitted. Before opening the file specified in this attribute, it shall be checked against the “type” attribute.

h_src Defines an image of the ruled line for horizontal writing. A pattern of an image is specified in the standard *Filename* format. This attribute takes precedence over both the “width” and “color_space, color, opacity” attributes. The viewer will repeatedly display the image. In this case, the viewer will display the defined file as is (without rotating) according to the writing direction even if the drawing area is cut off in the middle of a pattern. When the viewer recognizes the direction of writing automatically, only either the “v_src” or “h_src” attribute has to be defined. Note that the ruled line itself will be displayed vertically (for horizontal writing).
The PNG format and JPEG format image files are selectable in this standard. Cannot be omitted if the “type” attribute is specified. Nothing is set for both “src” and “type” attributes if they are omitted. Before opening the file specified in this attribute, it shall be checked against the “type” attribute.

type Stores the MIME type of the ruled image. The following formats are available in this standard. Cannot be omitted if the “src” attribute is specified.

PNG format:	"image/png"
JPEG format:	"image/jpeg"

[Child elements]
<permission_info>
Defines the permissions for the ruled image specified by the "src" attribute. Written in the standard *Permission* format. Can be omitted. If the “src” attribute does not specify the ruled image, this permission information should be ignored.



IEC 2801/13

B.4.7.5 Image object instance

Regular image file referred to by the <object> tag of the *text object instance* (see B.4.7.2.3). In this standard, it can either be a PNG file, a JPEG file, or a CCF file.

B.4.7.6 Sound object instance

Regular sound file referred to by the <sound_object_entry> tag, or in the <flip_animation> tag. In this standard, the compatible file formats are limited to MP3, AAC and SMAF, 3GPP, and MP4.

B.4.7.7 Animation object instance

The following possibilities for Flip animation <flip_animation> are available.

Defines an animation as a sequence of images changing at fixed time intervals. Written as an XML file, as follows.

Example:

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

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

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

[Attribute]

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

[Child elements]

<flip_animation_sound> Defines a sound object that is to be played simultaneously to the animation. 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 shall be checked against the type attribute.

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

[Attributes]

type: Defines the type of image file registered by the src attribute as a MIME type. Cannot be omitted. In the present standard, the following formats are accepted:

"image/png"

"image/jpeg"

"image/x-ccf"

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

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

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

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

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

Example:

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

B.4.7.8 Movie object instance

Regular movie file referred to by the <movie_object_entry> tag (see B.4.6.5). In this standard, only 3GPP2, 3GPP, and MP4 formats are valid.

B.4.7.9 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>
```

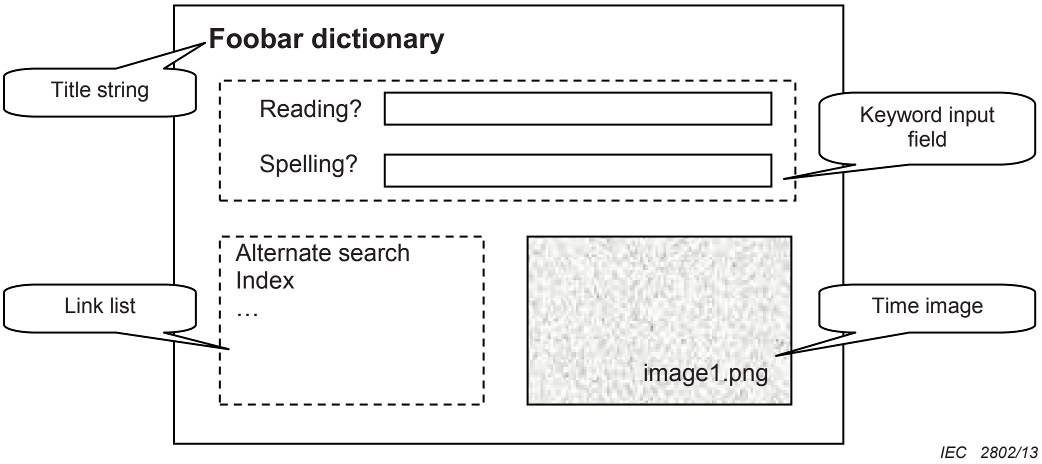
A rendering sample is also shown in Figure B.7. As can be seen, it is divided into 4 regions listed below. The way these regions are visually organized depends on the viewer. Some viewers may even decide not to display all of them.

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 string 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 2802/13

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 shall 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>
```

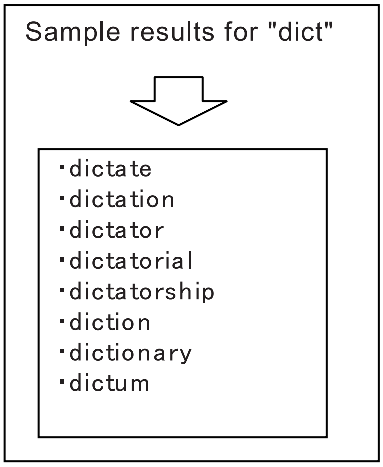
<key_input_region> Defines the keyword input region. Within a single search screen, there can only be one or two of instances of this element. The viewer uses this information to display the input region, and then conducts the search based on user input. It has the following attributes and child elements.

[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.4.4.2.6). 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"/"siborikomi" in that case.

"matches_only"/(alternatively)"siborikomi": "Matches only" search mode. Headwords that have common parts between the input character strings in the keyword input area and the top character are listed as a search result. The normalized (see B.4.4.2.6, <key_normalization>) keyword is forward-matched to each headword in the search table designated by the “table_id” attribute. The result of this search is all the words that match using those criteria, and only those words.

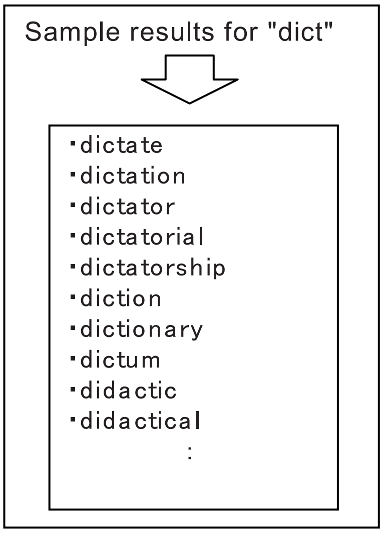


Sample results for "dict"

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

"matches_first"/(alternatively)"atama-dasi":

"Matches_first" mode. Headwords that have common parts between the input character strings in the keyword input area and the top character are listed as a search result. Headwords that are stored after the headword above are listed as well. The comparison method is the same as in "siborikomi", but the result list is a little different. It contains all the words matching the normalized keyword, and also the following entries in the search table.



Sample results for "dict"

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

:

[Child elements]

<key_input_region_prompt> Records the character string used as a label to explain the keyword input field, such as the "Reading?" or "Spelling?" shown in Figure B.7. Cannot be omitted.

[Child element]

The character string to be used as explained above.

<enable_key_type> Defines what types of characters are allowed in the keyword input field. Cannot be omitted.The attributes and child elements of this tag are the same as those used when it is in search table data’s <enable_key_type>(see B.4.4.2.6 for the details). The character types allowed in the search page object’s <enable_key_type> shall be a subset of those defined in the associated search table. Note that this means it is allowed for the former to prohibit character types enabled by the latter, whereas the opposite is not allowed.

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 specify other search tables, or a specific position in a text flow. The jumps are handled by using the <trigger_pointer> (see B.4.5.3) and <action_page_jump> (see B.4.5.4.3) tags. They should be recorded in the event data (see B.4.5.2) of the flow hosting this search page object. Can 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.

[Child element]

Combination of the extended character string and the <external_char> tag.

Example:

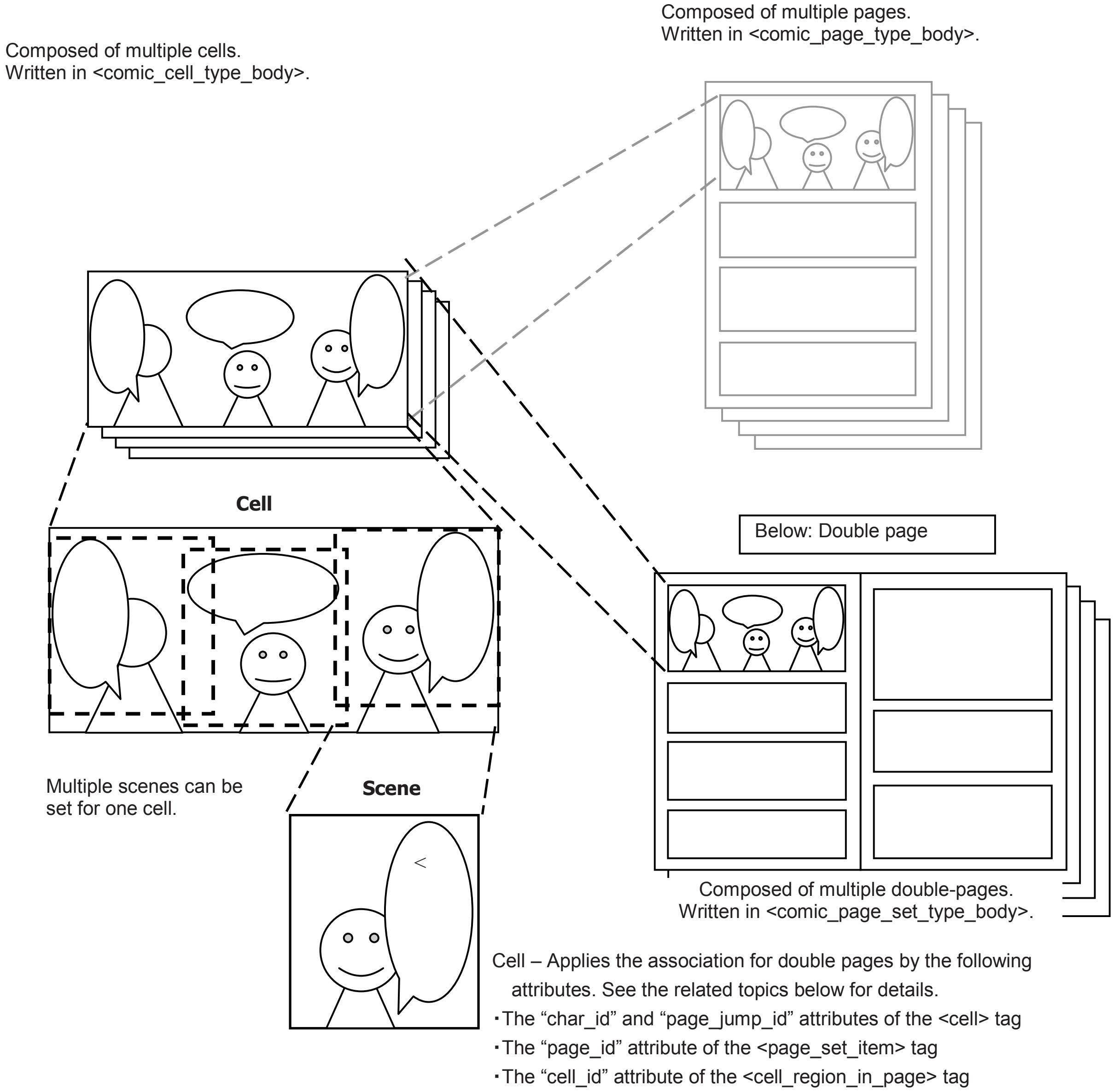
```
<search_link_item char_id="CR0001">
        <search_link_title>Alternate search</search_link_title>
</search_link_item>

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

B.4.7.10 Comic object instance <comic_data>

B.4.7.10.1 <comic_data> tag

Registers a *comic object instance*, whose structure is depicted in the following Figure B.8. As can be seen, the *comic object instance* is composed of multiple cells and multiple pages, and for each cell multiple scenes are set. In using a printed comic as an example, a “cell” on the paper page corresponds to the “cell” in this standard.


Composed of multiple cells.
Written in <comic_cell_type_body>.
Composed of multiple pages.
Written in <comic_page_type_body>.
Cell
Below: Double page
Multiple scenes can be set for one cell.
Scene
<
Composed of multiple double-pages.
Written in <comic_page_set_type_body>.

Cell – Applies the association for double pages by the following attributes. See the related topics below for details.

- The “char_id” and “page_jump_id” attributes of the <cell> tag
- The “page_id” attribute of the <page_set_item> tag
- The “cell_id” attribute of the <cell_region_in_page> tag

IEC 2803/13

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. Switching displays between cells/pages is possible, and the pages are displayed in accordance with the stored date.

How to divide data into *comic object instances* (cell counts or page counts to be used, etc.) 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 group 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.4.7.10.2 for details.

B.4.7.10.2 Comic content data <comic_body>

This field stores the comic content data as will actually be displayed. In this standard, the data is stored in the cell/scene structure or in the double-page display structure.

[Attributes] None

[Child elements]

<comic_cell_type_body> Serves as a main container for cell/scene structured data. Cannot be omitted. There shall be only one instance of this tag. See B.4.7.10.3 for details.

<comic_page_set_type_body> Serves as a main container for double-page display structured data. Can be omitted. There shall be only one instance of this tag. See B.4.7.10.7 for details.

B.4.7.10.3 Cell/scene structured content data <comic_cell_type_body>

This field stores the cell display comic content data. The following attributes and child elements are available.

[Attributes]

bg_color, bg_color_space, bg_opacity

Defines the default background color of this cell/scene structured 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.4.7.10.4). 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>. If none is defined, the behavior depends on the viewer, and may follow user defined preferences.

scroll_free Defines whether or not the user is permitted to scroll each cell in this cell/scene structured content. When permitted, “permit” should be specified. When omitted, scrolling is forbidden.

“permit”	Scrolling is permitted.

Magnify Defines whether or not the user is permitted to enlarge or reduce cells in this cell/scene structured content. When permitted, “permit” should be specified. When omitted, resizing is forbidden.

“permit”	Resizing is permitted.

[Child elements]
<comic_background_music>
Defines the background music to be played back when this comic content data for cell display is displayed. If this tag is omitted, no background music is set. This tag has the following attributes and child elements.

[Attributes]

type Stores the MIME type of the background music. The following formats are available in this standard. Cannot be omitted.

MP3 format:	“audio/mp3”
AAC format:	“audio/3gpp2”
SMAF format:	“application/x-smaf”
3GPP format:	“audio/3gpp”
MP4 format:	“audio/mp4”

src *Filename* of the background music. Written in the standard *Filename* format. The choices are limited to the following format image files under the current specifications: MP3 (with ".mp3" extension), AAC (with ".3g2" extension), SMAF (with ".mmf" extension), 3GPP (with ".3gp" extension), and MP4 (with ".mp4" extension). Cannot be omitted. Before opening the file specified in this attribute, it shall be checked against the “type” attribute.

loop Defines whether the background music specified in the “src” attribute is played back repeatedly. Select either of the values below. "no" is set if omitted. If the following “sync” attribute is defined “yes”, this attribute has to be “no”.

"yes"	Played back in a loop.
"no"	Is not played back in a loop.

sync Defines whether or not the cell display process is synchronized with background music defined by the “src” attribute of the <comic_background_music> tag. Select either of the values below. "no" is set if omitted. If the “loop” attribute (mentioned above) is defined “yes”, this attribute has to be “no”.

"yes"	The cell display is synchronized with the background music.
"no"	The cell display is not synchronized with the background music.

If “yes” is set, the scrolling speed from the scene to the next one should be specified by the “bgm_sync” attribute of the <cell_scene> tag (see B.4.7.10.5).

[Child elements]

<permission_info> Defines the permissions for the background music specified by the "src" attribute. Written in the standard *Permission* format. Can be omitted.

<cell> Stores information for each individual cell. This tag is stored for the number of cells. There shall be at least one instance of this tag. See B.4.7.10.4 for details.

B.4.7.10.4 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 this standard, only JPEG and PNG format image files are allowed. Cannot be omitted. Before opening the file specified in this attribute, it shall 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 of <cell> or <cell_scene/> tags (see B.4.7.10.4 and B.4.7.10.5).

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

PNG format:	"image/png"
JPEG format:	"image/jpeg"

char_id: Defines the ID number of this cell, written in the standard *Char_ID* format. This is used as a target for jumps from the viewer's menu or from link jumps in *text object instances*, also to make an association between this cell and a specific page in double-page display data. Cannot be omitted if the comic content data for double-page display (<comic_page_set_type_body> tag) exists.

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 settings defined in the corresponding attributes of the <comic_cell_type_body> tag (see B.4.7.10.3). 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>). 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 was set to "no".

"yes":	Permits Fit display.
"no":	Does not permit Fit display.

When the viewer is allowed to resize the image, the whole cell is not resized to fit but the scene-centered rectangle area of the defined display size is resized to fit into the viewer's actual screen. Note that even if the fitting setting is permitted by this attribute, it may not be permitted in the cases listed below.

- The scrolling time is set for any of the cells (on a 1 to 10 scale by the "scroll_time" attribute).
- The actual display size is wider than the defined display size.
- The viewer doesn't handle this fitting feature

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.

“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.

“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.

“wipe-right”: the new cell's image wipes in rightwards, starting from left to right, over the old cell's image.

“wipe-left”: the new cell's image wipes in leftwards, starting from right to left, over the old cell's image.

“wipe-up”: the new cell's image wipes in upwards, starting from bottom to top, over the old cell's image.

“wipe-down”: the new cell's image wipes in downwards, starting from top to bottom, over the old cell's image.

“wipe-vertical-in”: the new cell's image wipes in, starting from top/bottom to center, over the old cell's image.

“wipe-vertical-out”: the new cell's image wipes in, starting from center to top/bottom, over the old cell's image.

“wipe-horizontal-in”: the new cell's image wipes in rightwards/leftwards, starting from left/right to center, over the old cell's image.

“wipe-horizontal-out”: the new cell's image wipes in rightwards/leftwards, starting from center to right/left, over the old cell's image.

“wipe-center-in”: the new cell's image wipes in, starting from four corners to center, over the old cell's image.

“wipe-center-out”:the new cell's image wipes in, starting from center to four corners, over the old cell's image.

“randomblock”: the new cell’s image is displayed gradually and randomly in blocks, over the old cell’s image.

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. It defaults to 5 when omitted.

effect_pos: when the effect is set to 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, it defaults to “no”. The following values can be used.

“custom” sets a scene to the arbitrary coordinate position (specified by the “position” attribute of the <cell_scene /> tag) and scrolls between the defined scenes.

“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. It is recommended that the scroll_time attribute of <cell_scene> should be used instead.

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. It is recommended that the vibration attribute of <cell_scene> should be used instead.

display_vibration: specifies whether or not making the display vibrate when displaying this scene. The possible values are "on" and "off". If omitted, defaults to "off". Vibration and displaying the superimposed image (the scene when the superimposed image is not defined) should start at the same time. When backlight blinking is set to "on" by the "backlight" attribute (mentioned below), this setting is not accepted. It is recommended that the display_vibration attribute of <cell_scene> should be used instead.

"on": Makes vibration
"off": No vibration

vibration_direction: defines the direction of the display vibration. The possible values are "vertical" and "horizontal". If omitted, defaults to "vertical". It is recommended that the vibration_direction attribute of <cell_scene> should be used instead.

"vertical": Vertical direction
"horizontal": Horizontal direction

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. It is recommended that the vibration_time attribute of <cell_scene> should be used instead.

backlight: defines whether or not the backlight blinks when this scene is displayed. The possible values are "on" and "off". If omitted, defaults to "off". Blinking of the backlight and displaying the superimposed image (the scene when the superimposed image is not defined) should start at the same time. If the "display_vibration" attribute is set to "on", this setting is not accepted. It is recommended that the backlight attribute of <cell_scene> should be used instead.

"on": The backlight blinks.
"off": No blinking.

backlight_time: sets the number of times the backlight blinks on a 1 to 5 scale. If omitted, defaults to "3". The backlight blinking and the device vibration should start at the same time when both attributes are set. In that case, the vibration periods (set by the "vibration_time" attribute) is ignored. It is recommended that the backlight_time attribute of <cell_scene> should be used instead.

backlight_cycle: sets the blinking cycle on a 1 to 3 scale, 1 being the shortest, and 3 the longest. Note that the actual speed depends on the viewer and the device in use. It defaults to 2 when omitted. It is recommended that the backlight_cycle attribute of <cell_scene> should be used instead.

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 shall 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. It is recommended that the sound attribute of <cell_scene> should be used instead.

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. It is recommended that the sound_type attribute of <cell_scene> should be used instead.

"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. It is recommended that the url_jump of <cell_scene> should be used instead.

id defines the ID number of this cell. Must be uniquely given to cell data (<cell> tag) or page data (<page> tag) in a single instance of <comic_body>. This is used as a target for jumps from page-based display to cell-based. Using char_id attribute instead is recommended. Otherwise, cannot be omitted if the comic content data for double-page display (<comic_page_type_body> tag) exists.

page_jump_id: specifies a page related to this cell by the page ID number defined by the “page_id” attribute of the <page_set_item> tag. Written in the standard *Char_ID* format. Cannot be omitted if the comic content data for double-page display (<comic_page_set_type_body> tag) exists. This is used as a jump target when the cell display is changed to the page display.

[Child element]

<cell_scene> The area for storing each scene data information. The number of this <cell_scene> tag should be determined by the “scroll” attribute as follows.

When scroll=”custom”, it is necessary to specify this tag as the same number of the scenes.
When scroll=“up”, “down”, “left”, “right”, it is necessary to specify two <cell_scene> tags.
When scroll=”no”, it is necessary to specify one <cell_scene> tag.

Description example: in cell/scene structured comic content only

```
<comic_data>
        <!—- Content area -->
        <comic_body>
                <comic_cell_type_body bg_color="white">
                        <!—- The first cell -->
                         <cell     src="cell001.png"     type="image/png"     char_id="CR0001"
bg_color="blue" effect="fade-in" scroll="custom">
                                <cell_scene … />
                                <cell_scene  …  />
                                …
                        </cell>
                        <!—- The second cell -->
                        <cell     src="cell002.png"     type="image/png"     char_id="CR0002"
scroll="custom">
                                <cell_scene  …  />
                                 <cell_scene  …  />
                                …
                        </cell>
                        <!—- The third cell -->
                        <cell src="cell003.png" type="image/png" fit="yes" scroll="no" >
                                <cell_scene  …  />
                                </cell>
                        <cell src="cell004.png" type="image/png" scroll="up" />
                                <cell_scene  …  />
                                <cell_scene  …  />
                        </cell>
                        <!—- The fourth cell -->
        </comic_cell_type_body>
        </comic_body>
</comic_data>
```

Cell/scene structured comic content and double-page comic content are included.

```
<comic_data>
        <!—- Content area -->
        <comic_body>
                <comic_cell_type_body bg_color="white">
                        <!—- The first cell -->
                        <cell      src="cell01.png"      type="image/png"      scroll="custom"
char_id="CR0001" page_jump_id="CRpg01">
                                 <cell_scene  …  />
                                <cell_scene  …  />
                                …
                        </cell>
                        <!—- The second cell -->
                        <cell     src="cell02.png"     type="image/png"002"     scroll="custom"
char_id="CR0002" page_jump_id="CRpg01">
                                <cell_scene  …  />
                                 <cell_scene  …  />
                                …
                                </cell>
                                …
                        </comic_cell_type_body>
                <comic_page_set_type_body bg_color="white" … >
                        …
                 </comic_page_set_type_body>
        </comic_body>
</comic_data>
```

B.4.7.10.5 Cell scene data <cell_scene>

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

[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. Can be omitted. If the "scroll" attribute value of the <cell> tag is not "custom", this "position" attribute is ignored. If it is "custom and this "position" attribute is omitted, the viewer behaves as if the center point of the cell image is specified.

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. It defaults to 5 when omitted. If "0" is specified, the scrolling screen is not displayed. If this scene is the last one in the cell (the last one among the cells defined by <cell_scene/> tag in the <cell> tag), this attribute is ignored.

draw_image_time Determines the speed to display the superimposed image over the scene, on a 1 to 3 scale, 1 being the shortest, and 3 the longest. Note that the actual speed depends on the viewer and the device in use. When omitted, the scene and the superimposed image are displayed simultaneously. When the <cell_draw_image> tag is not defined as a child element for the scene, this setting 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, it defaults to "off". Vibration and displaying the superimposed image (the scene when the superimposed image is not defined) should start at the same time.

"on": Makes vibration
"off": No vibration

display_vibration: Specifies whether or not making the display vibrate when displaying this scene. The possible values are "on" and "off". If omitted, defaults to "off". Vibration and displaying the superimposed image (the scene when the superimposed image is not defined) should start at the same time. When backlight blinking is set to "on" by the "backlight" attribute (mentioned below), this setting is not accepted.

"on": Makes vibration
"off": No vibration

vibration_direction: Defines the direction of the display vibration. The possible values are "vertical" and "horizontal". If omitted, defaults to "vertical".

"vertical": Vertical direction
"horizontal": Horizontal direction

vibration_time: Sets the vibration periods for the device or the display on a 1 to 5 scale, 1 being the shortest, and 5 the longest. Note that the actual speed depends on the viewer and the device in use. It defaults to 3 when omitted.

backlight: Defines whether or not the backlight blinks when this scene is displayed. The possible values are "on" and "off". If omitted, defaults to "off". Blinking of the backlight and displaying the superimposed image (the scene when the superimposed image is not defined) should start at the same time. If the "display_vibration" attribute is set to "on", this setting is not accepted.

"on": The backlight blinks.

"off": No blinking.

backlight_time: Sets the number of times the backlight blinks on a 1 to 5 scale. If omitted, defaults to "3". The backlight blinking and the device vibration should start at the same time when both attributes are set. In that case, the vibration periods (set by the "vibration_time" attribute) is ignored.

backlight_cycle: Sets the blinking cycle on a 1 to 3 scale, 1 being the shortest, and 3 the longest. Note that the actual speed depends on the viewer and the device in use. It defaults to 2 when omitted.

sound: Designates a sound file to be played when this scene is displayed. Written in the standard *Filename* format. The choices are limited to the following format audio files in this standard: MP3 (with ".mp3" extension), AAC (with ".3g2" extension), SMAF (with ".mmf" extension), 3GPP (with ".3gp" extension), and MP4 (with ".mp4" extension). Before opening the file specified in this attribute, it shall be checked against the "sound_type" attribute. The sound playback and the display of the superimposed image (the scene when the superimposed image is not defined) should start at the same time, and the sound is not repeated. When the "sound_type" attribute is specified, this attribute cannot be omitted. If both "sound" and "sound_type" attributes are omitted, no sound is to be played.

sound_type: Stores the MIME type of the sound file. The choices are limited to the following formats in this standard. Cannot be omitted if the "sound" attribute is specified.

MP3 format:	"audio/mp3"
AAC format:	"audio/3gpp2"
SMAF format:	"application/x-smaf"
3GPP format:	"audio/3gpp"
MP4 format:	"audio/mp4"

url_jump: Associates the URL of a web site (limited to the html format) or a related point within the content with the current scene. The following formats are available. When omitted, handles as if nothing is associated with the scene.

(a) Writes the URL of a website.
Only URL beginning with "http://" and "https://" are allowed. Refer to the RFC 3986 for details about the format to use.

(b) Writes a related point within the content.
The following formats are possible depends on the point's position (whether the beginning or the middle of a flow).

- Specifies the beginning of a flow.
Writes the *Page ID* number of the flow data in the standard "local://*Page_ID*" format (i.e. url_jump="local://PG0001").
- Specifies the middle of a flow.
Defines the target of the link written as "url_jump=local://*Page_ID*/*Object_ID*/*Char_ID*" (i.e. url_jump="local://PG0001/OB0001/CR0001").
The *Page_ID* should be the ID number of the link target flow data.
The *Object_ID* should be the object ID number of the content data.
The *Char_ID* should be the ID number of a text string defined within the object.

bgm_sync: Sets the time (elapsed time after the playback is started) to move on to the next scene from the current one when the scene moves on to the next scene with the background music. Written in the standard Time format. Can be omitted. Note that if the "sync" attribute of the <comic_background_music> tag (in which this <cell_scene> tag is included) is set to "yes", this attribute cannot be omitted except when this is the last scene in the flow. If the "sync" attribute is set to "no", or this scene is the last scene in the flow even when the setting is "yes", this "bgm_sync" attribute is ignored.

[Child elements]

<cell_draw_image/>

The area for storing information on the superimposed image to be used over the cell's image. Plural settings allowed. Can be omitted. When two or more images are defined by this tag, the viewer displays the images at the same time. In that case, the display order is determined according to the recorded order when the images are overlaid with each other. In other words, the latter recorded images should be displayed over the previous one. The speed of the graphic overlay effect is determined by the "draw_image_time" attribute of the <cell_scene> tag. When the scene is scrolled during an animation effect (defined by the "effect" attribute of the <cell> tag), the viewer displays the superimposed image in the following manner.

- When scrolling the scene with the superimposed image to the next scene, the images remain displayed on the screen during scrolling. In that case, only the image overlaid over the previous scene should be displayed.

- During an animation effect, the viewer does not display the superimposed image of the cell's scene. The image is displayed after the animation effect is ended at the time defined by the "draw_image_time" attribute of the <cell_scene> tag. If the cell image remains displayed for the animation effect (i.e. "overwrite-right" or "pushout-left" is set by the "effect" attribute of the <cell> tag), the superimposed image over the scene also remains displayed.

B.4.7.10.6 Superimposed image data <cell_draw_image/>

The area for storing the information for the image superimposed on the cell images. This tag has the following attributes.

[Attributes]

src: Specifies the superimposed image to be displayed over the cell. For example, this is used when an enlarged image of a balloon part is overlaid over the cell. Written in the standard *Filename* format. The PNG format and JPEG format image files are selectable in this standard. Cannot be omitted. Before opening the file specified in this attribute, it shall be checked against the "type" attribute.

type: Stores the MIME type of the image to be specified by the "src" attribute. The choices are limited to the following formats in this standard. Cannot be omitted.

PNG format:	"image/png"
JPEG format:	"image/jpeg"

position: Specifies the point in the cell image where the superimposed image (defined by the "src" attribute) is displayed, in the standard Coordinates format. The viewer displays as the point defined by this attribute fits the upper-left apex of the superimposed image. Cannot be omitted.

Example: Without BGM settings

```
<cell src=”cell001.png” type=”image/png” char_id=”CR0001” bg_color=”blue”>
        <!—- The first scene -->
        <cell_scene    position=”    （   30,50   ）    ”    scroll_time=”1”    sound=”sound001.mp3”
sound_type=”audio/mp3”
                   backlight=”on” backlight_time=”5” backlight_cycle=”1”/>
        <!—- The second scene -->
        <cell_scene    position=”    （    100,200    ）    ”    vibration=”on”    vibration_time=”4”
url_jump=”http://www.sharp.co.jp/”
                   draw_image_time=”1”>
               <cell_draw_image src=”ballon001.png” type=”image/png” position=”（100,200）”/>
                 ・・・
          </cell_scene>
            ・・・
        </cell>
```

With BGM settings

```
<cell src=”cell001.png” type=”image/png” char_id=”CR0001” bg_color=”blue”>
        <!—- The first scene -->
   <cell_scene   position=”   （  30,50  ）   ”   scroll_time=”1”   backlight=”on”   backlight_time=”5”
backlight_cycle=”1”
              bgm_sync=”5s” />
        <!—- The second scene -->
              <cell_scene    position=”    （   100,200   ）    ”    vibration=”on”    vibration_time=”4”
url_jump=”http://www.sharp.co.jp/”
                   draw_image_time=”1” bgm_sync=”10s” >
               <cell_draw_image src=”ballon001.png” type=”image/png” position=”（100,200）”/>
                 ・・・
          </cell_scene>
            ・・・
        </cell>
```

B.4.7.10.7 Comic content data for double-page display <comic_page_set_type_body>

The area for storing the comic content data for double-page display. It has the following attributes and a child element.

[Attributes]

bg_color, bg_color_space, bg_opacity

Defines the default background color, in the standard *Color* format, of the image display area when displaying each page image in this comic content data for double-page display. This setting cannot be changed by the user. If the background color of the image display area for each double-page is defined (set by the “bg_color”, “bg_color_space” and the “bg_opacity” attributes of the <page_set> tag (see B.4.7.10.8)), the local settings take precedence. Can be omitted. When none of the <comic_page_set_type_body> and <page_set> tags are defined for the background color, the value set by the <flow_default_background> tag (in the <flow_default_attribute> ; see B.4.4.2.3) is used. If the <flow_default_background> tag is not defined, the behavior depends on the viewer, and may follow user defined preferences.

binding Defines a binding method for the page image in this comic content data for double-page display. It also determines This attribute determines the order in which the page images (defined by the <page_set_item> tag. See B.4.7.10.9) appear on the right or the left. The following values can be used: “right” is set if omitted.

“right”: Right binding. The first page image recorded in the <page_set_item> tags of the <page_set> tag is displayed on the right side, the second one on the left and so on.

"left": Left binding. The first page image recorded in the <page_set_item> tags of the <page_set> tag is displayed on the left side, the second one on the right and so on.

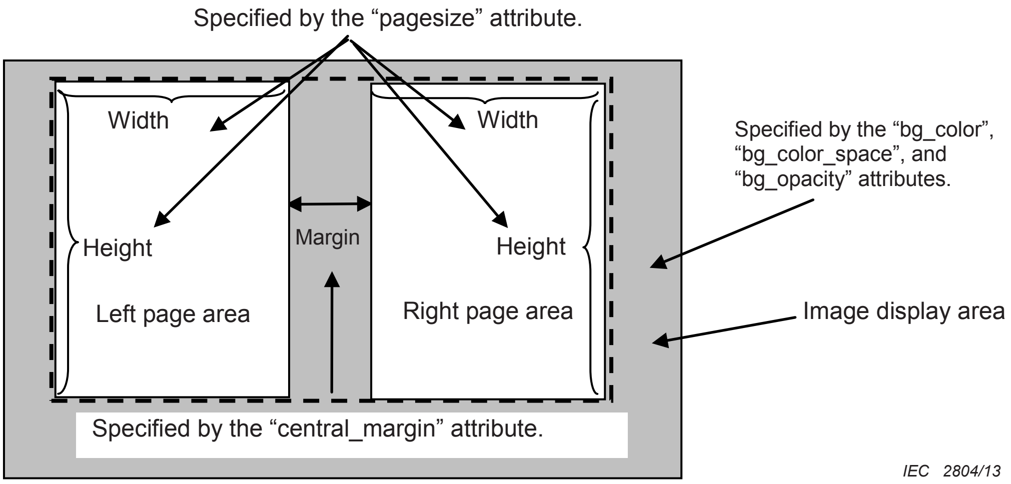
page_size Defines the default size of the page image for each left and right page display area in this comic content data for double-page display. Specified in the standard Coordinates format. If the size is defined for each double-page (set by the "page_size" attribute of the <page_set> tag (see B.4.7.10.8)), the local settings take precedence. Cannot be omitted.

central_margin Defines the default size, in number of pixels, of the width between the left and right page display areas (back margin width) when displaying each page image in this comic content data for double-page display. If the size is defined for each double-page (set by the "page_size" attribute of the <page_set> tag (see B.4.7.10.8)), the local settings take precedence. "0" is set if omitted.

[Child element]
<page_set> The area for storing information on each double-page data. Each double-page has to be defined by this tag. There shall be at least one. The viewer displays the page images in the recorded order of the double-page data.

The relation between the "page_size" and "central_margin" attributes is shown below.

The display area for each double page (right and left) and also the display area including the back margin part (the dot-lined rectangle part shown below) depend on the viewer.


Specified by the "pagesize" attribute.
Width
Width
Specified by the "bg_color", "bg_color_space", and "bg_opacity" attributes.
Height
Margin
Height
Left page area
Right page area
Image display area
Specified by the "central_margin" attribute.
IEC 2804/13

B.4.7.10.8 Double-page data <page_set>

The area for storing the double pages data. This tag has the following attributes and the child elements.

[Attributes]
type Defines the page image type of this double-page. The following values can be used: Can be omitted. When omitted, it is considered that the double-page is consisted of two page images (right and left). When this attribute is set as "spread", there shall be only one <page_set_item> tag recorded in the <page_set> tag. If this "type" attribute is omitted, it means there are two <page_set_item> tags (one for the right page, another for the left. See B.4.7.10.9).
"spread": Two page images combined together to one double-page image.

page_size Defines the size of the page display area for the page image of this double-page (defined by the <page_set_item> tag. See B.4.7.10.9), written in the standard Coordinates format. This attribute is set only when the size of this double-page display area is different from the default value. When the "type" attribute is set as "spread" in the <page_set> tag, this attribute cannot be omitted. If not set as "spread", can be omitted, and it defaults to the value defined by the <page_size> attribute of the <comic_page_set_type_body> tag.

central_margin Defines the back margin width to display the page image of this double-page (defined by the <page_set_item> tag. See B.4.7.10.9), written in number of pixels. This attribute is set only when the back margin width of this double-page is different from the default value. Can be omitted. It defaults to the value defined by the "central_margin" attribute of the <comic_page_set_type_body> tag when omitted. When the "type" attribute is set as "spread" in the <page_set> tag, this attribute is ignored.

bg_color, bg_color_space, bg_opacity
Defines the background color of the page display area for the page image of this double-page (defined by the <page_set_item> tag. See B.4.7.10.9), written in the standard *Color* format. This setting cannot be changed by the user. This attribute is set only when the background color of this double-page is different from the default value. Can be omitted. When omitted, the setting values for the comic content data for double-page display (specified by the "bg_color" attribute, "bg_color_space" attribute, and the "bg_opacity" attribute of the <comic_page_set_type_body> tag) are used.
When none of the <comic_page_set_type_body> and <page_set> tags are defined for the background color, the value set by the <flow_default_background> tag (in the <flow_default_attribute>. See B.4.4.2.3) is used. If the <flow_default_background> tag is not defined, the behavior depends on the viewer, and may follow user defined preferences.

[Child elements]
<page_set_item> The area for storing information on each page data. When the "type" attribute is set as "spread" in the <page_set> tag, there shall be only one <page_set_item> tag. When the "type" attribute is omitted, there shall be two <page_set_item> tags. When the "binding" attribute is set as "right" in the <comic_page_set_type_body> tag, the first <page_set_item> tag should be for the right page, the second for the left. In contrast, when the "binding" attribute is set as "left", the first <page_set_item> tag is for the left page, and the second for the right.

B.4.7.10.9 Unilateral page data <page_set_item>

The area for storing each page data to make up the double pages. This tag has the following attributes and the child elements.

[Attributes]
src Defines the filename for the displayed page image. Written in the standard *Filename* format. The PNG format and JPEG format image files are selectable in this standard. A blank page is set if omitted. Before opening the file specified in this attribute, it shall be checked against the "type" attribute.

type Sets the image file type to be specified by the "src" attribute using the MIME format. The choices are limited to the following formats under the current specifications. Cannot be omitted if the "src" attribute is specified.

PNG format:	"image/png"
JPEG format:	"image/jpeg"

position Specifies the position where the page image is displayed in the display area (defined by the "page_size" attribute of the <comic_page_set_type_body> tag and the “page_size” attribute of the <page_set> tag). Writes the point where the upper-left apex of the page image (defined by the “src” attribute) is placed, in the standard Coordinates format. Can be omitted, defaults to “x=0, y=0” meaning the upper-left corner of the display area.

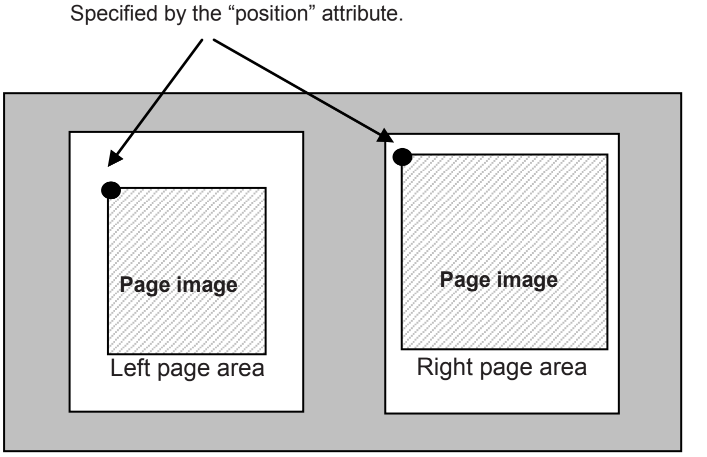
page_id Defines the page ID to be set for this page. Written in the standard *Char_ID* format. Can be omitted. No ID setting if omitted.

[Child elements]
<cell_region_in_page>

Stores information of each cell in this page by this tag. It has the following attributes and child elements. Cannot be omitted, when the “src” and “type” attributes of the <page_set_item> are defined.

[Attributes]
cell_id Sets the ID number of the cell related to this area in the standard *Char_ID* format. Cannot be omitted. Records the ID number defined by the “char_id” attribute of the <cell> tag.

The relation between the “position” attribute and the page display area (defined by the “page_size” attribute of the <comic_page_set_type_body> tag or the “page_size” attribute of the <page_set> tag) is shown below. Note that the position to display the page display area in the screen depends on the viewer.



IEC 2805/13

Example:

```
<comic_page_set_type_body binding=”right” page_size=”(600,800)” central_margin=”20”
              bg_color=”white”>
    <page_set>
    <page_set_item/>
    <page_set_item src=”page01.png” page_id=”CRpg01” bgm_sync= ”5s” >
      <cell_region_in_page cell_id=”CR0001”/>
      <cell_region_in_page cell_id=”CR0002”/>
       …
    <page_set_item/>
  </page_set>
  <page_set>
    <page_set_item src=”page02.png” position=”(10,10)” page_id=”CRpg02” >
      <cell_region_in_page cell_id=”CR0010”>
      <cell_region_in_page />
       …
    <page_set_item src=”page03.png” page_id=”CRpg03” >
      <cell_region_in_page cell_id=”CR0020”/>
       …
    <page_set_item/>
  </page_set>
  <page_set type=”spread” page_size=”(1000,800)”>
    <page_set_item src=”page04.png” page_id=”CRpg04” >
                <cell_region_in_page cell_id=”CR0030”/>
       …
    <page_set_item/>
  </page_set>
  <page_set>
    …
  </page_set>
  …
</comic_page_set_type_body>
```

B.4.7.11 Dictionary data object instance <dict_data>

B.4.7.11.1 <dict_data> tag

Dictionary data object instance <dict_data> has the following child elements.

[Child elements]
<dict_default_attribute>
Dictionary data default attribute area (see B.4.7.11.2). Cannot be omitted.

<dict_body>
Dictionary data content area (see B.4.7.11.3). Cannot be omitted.

B.4.7.11.2 Dictionary data default attribute area <dict_default_attribute>

The area to record a display parameter to be used when displaying this *dictionary data object instance*. The values or the methods set in this tag will serve as default, it may proceed without respecting the value defined in these default value tags when the viewer does not implement the required method, or wishes to give priority to user settings, except when the following explanations state otherwise. In addition, some of the parameters defined here may also be set in the whole text flow. These are used as default if the local ones (the ones defined here) are not set when the object defined in the whole text flow is the dictionary data object. In case both are defined, the local settings take precedence.

Dictionary data default attribute area <dict_default_attribute> tag has the following attributes and child elements.

[Attributes]

Baseline Defines the baseline direction for displaying this *dictionary data object instance*, in either attribute option below. Not specified if this tag is omitted.

“right“ Horizontal writing direction (from left to right) is recommended. However, the direction can be changed according to the user preferences.

“right-only“ The only possible option is a horizontal writing (from 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“ Vertical writing direction (from top to bottom) is recommended. However, the direction can be changed according to the user preferences.

“down-only“The only possible option is a vertical writing (from 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 the method to decide the position of *dictionary data object instance* according to the line-direction (i.e. horizontal position for vertical writing). The following value is eligible. When omitted, the object should be displayed from the top of the display area.

“middle“ The content in the <dict_body> tag is centered within the display area. If the content is larger than the display area, this attribute is ignored.

[Child elements]

<dict_default_font/>

Defines the default font and font color for this text object. When omitted, it defaults to the <flow_default_attribute> value.
This tag has the following attributes.

[Attributes]

fontname Default font name. More than one fonts 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, it defaults to the <flow_default_attribute> value.

color_space,color,opacity Defines the default character color, in the standard *Color* format. Not specified if the “color” attribute is omitted.

ruby_flag Defines whether ruby in the content data is to be viewed or not. The following values can be used: Not specified if this tag is omitted. If set to “yes” or “yes_only”, ruby is displayed for the character strings sandwiched by the <ruby> tags in the *dictionary data object instance*.

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

“yes_only“ Ruby shall be displayed. This option cannot be turned off by the user. However, this does not apply to viewers that are not able to display ruby.

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

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

<dict_default_background>

Defines the background color and background image to be used for this *object instance*. When both are defined, the image is drawn centered in the screen filled with the background color. Finally, the content of the <dict_body> tag is rendered on the top. The behavior in case of omission of this tag should conform to the behavior specified in case of omission of its attributes.

[Attributes]

color_space,color,opacity Defines the default background color of the display, in the standard *Color* format. If the "color" attribute is omitted, none is specified.

type Stores the MIME type of the background image. The following formats are available in this standard. Cannot be omitted if the "src" attribute is specified.

PNG format:	"image/png"
JPEG format:	"image/jpeg"
CCF format:	"image/x-ccf"

src *Filename* for the background image. Written in the standard *Filename* format. The PNG format, JPEG format, and CCF format image files are settable for this attribute in this standard. Cannot be omitted if the "type" attribute is specified. Nothing is set for both "src" and "type" attributes if they are omitted. Before opening the file specified in this attribute, it shall be checked against the "type" attribute.

[Child elements]

<permission_info>

Defines the permissions for the background image specified by the "src" attribute. Written in the standard *Permission* format. Can be omitted. If the "src" attribute does not specify the background image, this permission information should be ignored.

<dict_default_background_music>

Defines the background music to be played back when this *object instance* is displayed. If this tag is omitted, no background music is set.
This tag has the following attributes and child elements.

[Attributes]

type Stores the MIME type of the background music. The following formats are available in this standard. Cannot be omitted.

MP3 format:	"audio/mp3"
AAC format:	"audio/3gpp2"
SMAF format:	"application/x-smaf"
3GPP format:	"audio/3gpp"
MP4 format:	"audio/mp4"

src *Filename* of the background music. Written in the standard *Filename* format. The choices are limited to the following format audio files in this standard: MP3 (with ".mp3" extension), AAC (with ".3g2" extension), SMAF (with ".mmf" extension), 3GPP (with ".3gp" extension), and MP4 (with ".mp4" extension). Cannot be omitted. Before opening the file specified in this attribute, it shall be checked against the "type" attribute.

loop Defines whether the background music specified by the “src” attribute should be played back repeatedly. Specified by either of the following attribute options. “no” is set if omitted.

“yes” Played back in a loop.
“no” Is not played back in a loop.

[Child elements]
<permission_info>
Defines the permissions for the background music specified by the "src" attribute. Written in the standard *Permission* format. Can be omitted.

<line_breaking_method>/<kinsoku> Defines the recommended line breaking method when this *object instance* is displayed. Can be omitted. This tag has the following attributes and child elements.

[Attributes]
method Defines the line breaking processing. Specified by either of the following attribute options. Not specified if this tag is omitted.

“none“ No line-breaking applied.
“run down“ Line-breaking applied (characters that shall not appear at the beginning/end of a line are driven to the previous/next line)

hanging_punctuation Defines whether processing of hanging characters is activated when a target character exists at the end of the line. Specified by either of the following attribute options. Not specified if this tag is omitted.

“yes” Processing of hanging characters is activated.
“no” Processing of hanging characters is disabled.

[Child elements]
<top_prohibit_char>
Stores the characters that shall not appear at the beginning of a line. Listed as an extended character string in the “、。.,” format. Not specified if this tag is omitted.

<end_prohibit_char>
Stores the characters that shall not appear at the end of a line. Listed as an extended character string in the “、。.,” format. Not specified if this tag is omitted.

<hanging_char>
Stores the characters to be displayed as hanging punctuation. Listed as an extended character string in the “、。.,” format. Not specified if this tag is omitted.

B.4.7.11.3 Dictionary data content area <dict_body>

The area for storing the dictionary data content. This tag has the following child elements.

[Child elements]
<word> Stores information for each word. There is one instance of this tag for each word. There shall be one or more instances of this tag.

B.4.7.11.4 Word data <word>

Stores dictionary information (i.e. headwords or definitions) for each word. It has the following attributes and child elements.

[Attributes]

page_break Defines whether or not a page break is inserted at the end of this <word> tag. Accepts the values listed below. “no” is set if omitted.

“yes”	Inserts a page break.
“no”	Does not insert a page break.

turning_page_control Defines how ordinary scrolling interacts with the page breaks, by allowing or forbidding crossing the page break forward or backward, in the case the “page_break” attribute is set to “yes”. The following options are available.

“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 overall setting defined by the “turning_page_control” attribute of the <flow_data> tag).

head_button_control Defines the restriction of how the viewer behaves when the keys for moving to the next/previous headword (i.e. the beginning of the definitions) is pushed under the condition of the “page_break” attribute set to “yes”. The following options are available.

“on”	Moving both to the next or previous data is forbidden.
“off”	Moving to the next and previous data 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.

When omitted, it defaults to the “head_botton_control” attribute of <flow_data> tag inside a flowing content text object of this *object instance*.

revision Specifies the version for the revision in which the <word> tag appeared as an addition. For example, Takes value “1” for the first revision and ‘2” for the second. Can be omitted if there is no such revision information, usually meaning the content data is for the first version.

delete Defines whether or not to delete the content of this <word> tag. Accepts the values listed below. It defaults to “no” when omitted. This attribute value is used when the contents is revised. However, special attention should be paid, if a word has been deleted (this attribute value has been set to “yes” for the word) in one revised version, and then wishes to re-add the word in another version. In that case, the <word> tag has to be newly added to define the “revision” attribute instead of just defining the “revision” attribute to the <word> tag in which “yes” had been set to the “delete” attribute (in the previous version).

“yes”	Deletes the content.
“no”	Does not delete the content.

[Child elements]

The following tags are eligible for the <word> tag. See B.4.7.2.3 for details of each tag.

<p>, <br/>, <hr/>, <font>, <ruby>, <external_char>, <mask>, <char_id>, <key_entry>, <object>, <scrolling_text>(<telop>), <holizontal>(<yoko>)

Note that the following instructions should be followed for the <key_entry> tag (used to define headword information for searching capabilities) and the <key_item> tag.

There shall be one <key_entry> tag recorded in the beginning of the <word> tag.
There shall be one or more <key_item> tags in the <key_entry> tag. When there are two or more <key_item> tags, the information of the first recorded one is used for the bookmark functions or the wordbook functions for the dictionary.
The <external_char> tag cannot be used for the child element of the <key_item> tag.

Example:

```
<dict_data>
  <dict_default_attribute>
      :
  </dict_default_attribute>
 <dict_body>
   <word page_break=“yes” turning_page_control=“off”
head_button_control=“off”>
     <key_entry>
       <key_item table_id=“ST0001” search_word=“Mountain”>Mountain [mountain]
</key_item>
       <key_item table_id=“ST0001” search_word=“Mountain”>Mountain [mountain]
</key_item>
     </key_entry>
     <!-- What follows is what will be displayed as the result of the search -->
     <font color=”red” size=”150%” bold=”yes”> Mountain [mountain] </font><br/>
    Higher than the flatlands...<br/>
         :
     [1]...
     [2]...
   </word>
   <word ...  revision=“2”>  ...  </word>
   <word ... revision=“1” delete=“yes”>  ...  </word>
       :
  </dict_body>
</dict_data>
```

B.5 Color names (normative)

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	#FFFFF0	#FFFACD	#FFE4B5	

B.6 Localization (informative)

B.6.1 Possible additions

The proposals given here for localization are 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, examples of character sets for different languages are listed in Table B.7.

Table B.7 – Examples of standard character sets for different languages

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

B.6.3 Characters usable for reading

Examples of the language-specific character sets for reading are listed in Table B.8.

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

Target language	Name	Description
Japanese	Full-width Japanese katakana characters	ァアィイゥウェエォオカガキギクグケゲコゴサザシジスズセゼソゾタダチヂッツヅテデトドナニヌネノハバパヒビピフブプヘベペホボポマミムメモャヤュユョヨラリルレロヮワヰヱヲンヴヵヶ (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) Î (0x00CE) Ï (0x00CF) Ô (0x00D4) Ö (0x00D6) Ù (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 (0x0FF) Œ(0x0152) œ (0x0153)

B.6.4 Sorting rules for <search_table_def>

The methods listed in Table B.9 are examples of additional sorting rules for different languages.

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 Æ B C Ç E É È Ê Ë F G H I Î Ï J K L M N O Ô Ö ŒP Q R S T U Ù Û 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.6.5 Additional attributes for <enable_key_type>

Additional language-specific extension attributes of <enable_key_type> are shown in Table B.10 in addition to those described in B.4.7.2.3. The manner in which they are supposed to be used is the same as the one listed in B.4.7.2.3.

Table B.10 – Example of additional language specific attributes for <enable_key_type>

French	French_alphabet _extensions	À (0x00C0)	Â (0x00C2)	Æ (0x00C6)	Ç (0x00C7)
		È (0x00C8)	É (0x00C9)	Ê (0x00CA)	Ë (0x00CB)
		Î (0x00CE)	Ï (0x00CF)	Ô (0x00D4)	Ö (0x00D6)
		Ù (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 (0x0FF)	Œ(0x0152)
		œ (0x0153)			

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

B.7.1 General

The syntax of the XMDF XML format is formally given here as compact RELAX NG schema data. Note that they are not necessarily optimized as RELAX NG schemas.

B.7.2 Tags and attributes for object instances, bibliographic data and other data for content structure

```
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? | zipcode? ) ,

    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
zipcode = element zipcode { zipcode_attlist, text }
zipcode_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" }?
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 &=
  [ a:defaultValue = "1.00" ] attribute bvf_version { text }?,
  attribute id_type { text }?,
  attribute id { text }?,
  attribute default_ccs { text },
  [ a:defaultValue = "bvf_flow:3.00D" ]
  attribute sub_version { string "bvf_flow:3.00D" }?,
  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 &= attribute structured { "no" }?
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_kinsoku? )
  }
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 { text }?,
  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_kinsoku_attlist,
    top_prohibit_char?,
    end_prohibit_char?,
    hanging_char?
  }
flow_default_kinsoku_attlist &=
  [ a:defaultValue = "none" ] attribute method { "none" | "run_down" }?,
  [ a:defaultValue = "yes" ] attribute hanging_punctuation { Yes_No }?
flow_default_kinsoku =
  element flow_default_kinsoku {
    flow_default_kinsoku_attlist,
    top_prohibit_char?,
    end_prohibit_char?,
    hanging_char?
  }
top_prohibit_char =
  element top_prohibit_char { top_prohibit_char_attlist, text }
top_prohibit_char_attlist &= empty
end_prohibit_char =
  element end_prohibit_char { end_prohibit_char_attlist, text }
end_prohibit_char_attlist &= empty
hanging_char = element hanging_char { hanging_char_attlist, text }
hanging_char_attlist &= empty
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 }?,
  [ a:defaultValue = "on" ]
  attribute head_button_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"
    | "flow_title"
  }?,
  attribute title { text }?
search_table =
  element search_table { search_table_attlist, search_table_def+ }
search_table_attlist &=
  [ a:defaultValue = "no" ] attribute bookmark { Yes_No }?,
  [ a:defaultValue = "no" ] attribute wordbook { Yes_No }?,
  [ a:defaultValue = "yes" ] attribute jump_search_root { Yes_No }?,
  [ a:defaultValue = "yes" ] attribute jump_search { Yes_No }?,
  [ a:defaultValue = "yes" ] attribute all_search { Yes_No }?
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 }?,
  attribute name { text }?,
  [ a:defaultValue = "otherunicode" ] attribute sorting_rule { "otherunicode" | "implicit" | "unicode" | "other"}?,
  attribute short_name { text }?,
  [ a:defaultValue = "no" ] attribute wild { Yes_No }?,
  [ a:defaultValue = "no" ] attribute blank { Yes_No }?,
  [ a:defaultValue = "no" ] attribute end { Yes_No }?,
  attribute help_page_id { text }?
enable_key_type =
  element enable_key_type { enable_key_type_attlist, empty }
enable_key_type_attlist &=
  [ a:defaultValue = "no" ] ( attribute numerals { Yes_No }? | attribute number { Yes_No }? ),
  [ a:defaultValue = "no" ] ( attribute basic_alphabet { Yes_No }? | attribute alphabet { Yes_No }? ),
  [ a:defaultValue = "no" ] attribute kana { Yes_No }?,
  [ a:defaultValue = "no" ] attribute kanji { Yes_No }?
key_normalization =
  element key_normalization { key_normalization_attlist, empty }
key_normalization_attlist &=
  [ a:defaultValue = "yes"] attribute capitalization {Yes_No}?,
  [ a:defaultValue = "no"]  attribute diactritic_removal {Yes_No}?,
  [ a:defaultValue = "delete" ]
  attribute cho_on { "delete" | "repeat" | "no" }?,
  [ a:defaultValue = "yes" ] attribute daku_on { Yes_No }?,
  [ a:defaultValue = "yes" ] attribute handaku_on { Yes_No }?,
  [ a:defaultValue = "yes" ] attribute soku_on { Yes_No }?,
  [ a:defaultValue = "yes" ] attribute yo_on { Yes_No }?,
  [ a:defaultValue = "yes" ] attribute other_small_kana { 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
     | dict_data_object_entry
     | csf_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 layout_info_src { text }?,
  attribute layout_info_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 }
csf_object_entry =
  element csf_object_entry {
    csf_object_entry_attlist, permission_info?, protection?, caption?
  }
csf_object_entry_attlist &=
  attribute src { text },
  attribute type { text },
  attribute object_id { text }

caption = element caption { caption_attlist, TextWithGaiji }
caption_attlist &= empty

protection = element protection { protection_attlist, empty }
protection_attlist &= empty
search_page_object_entry =
  element search_page_object_entry {
    search_page_object_entry_attlist, permission_info?
  }
```

```
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, empty }
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?
  }
comic_object_entry_attlist &=
  attribute src { text },
  attribute type { text },
  attribute object_id { text }
dict_data_object_entry =
  element dict_data_object_entry {
    dict_data_object_entry_attlist, permission_info?
  }
dict_data_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? | kinsoku? )
  }
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 type { text }?,
  attribute src { text }?

text_default_background_music =
  element text_default_background_music {
    text_default_background_music_attlist, permission_info?
  }
text_default_background_music_attlist &=
  attribute type { text },
  attribute src { text },
  [ a:defaultValue = "no" ] attribute loop { Yes_No }?

line_breaking_method_attlist &=
  attribute method { "none" | "run_down" }?,
  attribute hanging_punctuation { Yes_No }?
line_breaking_method =
  element line_breaking_method {
    line_breaking_method_attlist,
    top_prohibit_char?,
    end_prohibit_char?,
    hanging_char?
  }
kinsoku_attlist &=
  attribute method { "none" | "run_down" }?,
  attribute hanging_punctuation { Yes_No }?,
  attribute alignment { "left" }?
kinsoku =
  element kinsoku {
    kinsoku_attlist,
    top_prohibit_char?,
    end_prohibit_char?,
    hanging_char?
  }

Block = p | ( scrolling_text | telop ) | key_entry
Inline =
  br
  | hr
  | font
  | ( yoko | horizontal )
  | ruby
  | external_char
  | mask
  | char_id
  | meaning_start
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" }?,
  attribute [ a:defaultValue = "1" ] drop_cap { text }?
scrolling_text =
  element scrolling_text {
    scrolling_text_attlist, (text | external_char | font | ( horizontal | yoko ) | ruby | object)*
  }
scrolling_text_attlist &= empty
telop =
  element telop {
    telop_attlist, (text | external_char | font | ( horizontal | yoko ) | ruby | object)*
  }
telop_attlist &= empty
page_break = element page_break { page_break_attlist, empty }
page_break_attlist &=
  attribute turning_page_control { Turn_Page_Val }?,
  attribute head_button_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

yoko = element yoko { yoko_attlist, TextWithGaiji }

yoko_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 | yoko ) | 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 }
meaning_start = element meaning_start { meaning_start_attlist, empty }
meaning_start_attlist &= empty
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" | "atama-dasi" ) | ( "matches_only" | "siborikomi" ) }?

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_page_set_type_body?,
comic_page_type_body?
  }
comic_body_attlist &= empty
comic_cell_type_body =
  element comic_cell_type_body {
    comic_cell_type_body_attlist, comic_background_music?, cell+
  }
comic_cell_type_body_attlist &=
  attribute bg_color_space { "RGB" }?,
  attribute bg_opacity { "100" }?,
  attribute bg_color { text }?,
  attribute scroll_free { "permit" }?,
  attribute magnify { "permit" }?
comic_background_music =
  element comic_background_music {
    comic_background_music_attlist, permission_info?
  }
comic_background_music_attlist &=
  attribute type { text },
  attribute src { text },
  [ a:defaultValue = "no" ] attribute loop { Yes_No }?,
  [ a:defaultValue = "no" ] attribute sync { Yes_No }?
cell = element cell { cell_attlist, cell_scene* }
cell_attlist &=
  attribute src { text },
  attribute type { text },
  attribute char_id { text }?,
  attribute id{ text }?,
  attribute bg_color_space { "RGB" }?,
  attribute bg_opacity { "100" }?,
  attribute bg_color { text }?,
  [ 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"
    | "box-rightdown"
    | "box-leftdown"
    | "box-rightup"
    | "box-leftup"
    | "box-center"
    | "box-free"
    | "pushout-right"
    | "pushout-left"
    | "pushout-up"
    | "pushout-down"
    | "wipe-right"
    | "wipe-left"
    | "wipe-up"
    | "wipe-down"
    | "wipe-vertical-in"
    | "wipe-vertical-out"
    | "wipe-horizontal-in"
    | "wipe-horizontal-out"
    | "wipe-center-in"
    | "wipe-center-out"
```

```
    | "randomblock"
  }?,
  attribute effect_time { text }?,
  attribute effect_pos { text }?,
  attribute scroll {
    "custom" | "up" | "down" | "left" | "right" | "no"
  }?,
  attribute page_jump_id { text }?,


  attribute scroll_time { text }?,
  attribute vibration { "on" | "off" }?,
  attribute display_vibration { "on" | "off" }?,
  attribute vibration_direction { "vertical" | "horizontal" }?,
  attribute vibration_time { text }?,
  attribute sound { text }?,
  attribute sound_type { text }?,
  attribute url_jump { text }?,
  attribute backlight { "on" | "off" }?,
  attribute backlight_time { text }?,
  attribute backlight_cycle { text }?
cell_scene = element cell_scene { cell_scene_attlist, cell_draw_image* }
cell_scene_attlist &=
  attribute position { text }?,
  attribute scroll_time { text }?,
  attribute vibration { "on" | "off" }?,
  attribute display_vibration { "on" | "off" }?,
  attribute vibration_direction { "vertical" | "horizontal" }?,
  attribute vibration_time { text }?,
  attribute sound { text }?,
  attribute sound_type { text }?,
  attribute url_jump { text }?,
  attribute backlight { "on" | "off" }?,
  attribute backlight_time { text }?,
  attribute backlight_cycle { text }?,
  attribute draw_image_time { text }?,
  attribute bgm_sync { text }?
cell_draw_image =
  element cell_draw_image { cell_draw_image_attlist, empty }
cell_draw_image_attlist &=
  attribute src { text },
  attribute type { text },
  attribute position { text }
comic_page_set_type_body =
  element comic_page_set_type_body {
    comic_page_set_type_body_attlist, page_set+
  }
comic_page_set_type_body_attlist &=
  [ a:defaultValue = "right" ] attribute binding { "left" | "right" }?,
  attribute page_size { text },
  [ a:defaultValue = "0" ] attribute central_margin { text }?,
  attribute bg_color_space { "RGB" }?,
  attribute bg_opacity { "100" }?,
  attribute bg_color { text }?
page_set = element page_set { page_set_attlist, page_set_item+ }
page_set_attlist &=
  attribute type { "spread" }?,
  attribute page_size { text }?,
  attribute central_margin { text }?,
  attribute bg_color_space { "RGB" }?,
  attribute bg_opacity { "100" }?,
  attribute bg_color { text }?
page_set_item =
  element page_set_item { page_set_item_attlist, cell_region_in_page* }
page_set_item_attlist &=
  attribute src { text }?,
  attribute type { text }?,
  [ a:defaultValue = "(0,0)" ] attribute position { text }?,
  attribute page_id { text }?
cell_region_in_page =
  element cell_region_in_page { cell_region_in_page_attlist, empty }
cell_region_in_page_attlist &=
  attribute cell_id { text },
  attribute region { text }?
comic_page_type_body =
  element comic_page_type_body { comic_page_type_body_attlist, page+ }
comic_page_type_body_attlist &=
  attribute bg_color_space { "RGB" }?,
  attribute bg_opacity { "100" }?,
  attribute bg_color { text }?,
  attribute order { "RT-LB" | "LT-RB" }?
page = element page { page_attlist, page_image+ }
page_attlist &=
  attribute bg_color_space { "RGB" }?,
  attribute bg_opacity { "100" }?,
  attribute bg_color { text }?,
  attribute id { text },
  attribute cell_jump_id { text }
page_image = element page_image { page_image_attlist, empty }
page_image_attlist &=
  attribute src { text },
  attribute type { text }

start = flip_animation | bvf | comic_data | text_data | search_page
```

B.7.3 Tags and attributes for dictionary data object instance

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

Yes_No = "yes" | "no"
```

```
TextWithGaiji = (text | external_char)*

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

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

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

permission_info =

  element permission_info {

    attlist_permission_info, print_permission?, copy_permission?

  }

attlist_permission_info &= empty

print_permission =

  element print_permission { attlist_print_permission, empty }

attlist_print_permission &=

  [ a:defaultValue = "no" ]

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

copy_permission =

  element copy_permission { attlist_copy_permission, empty }

attlist_copy_permission &=

  [ a:defaultValue = "no" ]

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

dict_data =

  element dict_data {

    attlist_dict_data, dict_default_attribute, dict_body

  }

attlist_dict_data &= empty

dict_default_attribute =

  element dict_default_attribute {

    attlist_dict_default_attribute,

    dict_default_font?,

    dict_default_background?,

    dict_default_background_music?,

    ( line_breaking_method? , kinsoku? )

  }
```

```
attlist_dict_default_attribute &=
  attribute baseline { BaseLine }?,
  attribute valign { "middle" }?
dict_default_font =
  element dict_default_font { attlist_dict_default_font, empty }
attlist_dict_default_font &=
  attribute fontname { text }?,
  attribute color_space { "RGB" }?,
  attribute opacity { "100" }?,
  attribute color { text }?,
  attribute ruby_flag { "yes" | "yes_only" | "no" | "no_only" }?
dict_default_background =
  element dict_default_background {
    attlist_dict_default_background, permission_info?
  }
attlist_dict_default_background &=
  attribute color_space { "RGB" }?,
  attribute opacity { "100" }?,
  attribute color { text }?,
  attribute type { text }?,
  attribute src { text }?

dict_default_background_music =
  element dict_default_background_music {
    attlist_dict_default_background_music, permission_info?
  }
attlist_dict_default_background_music &=
  attribute type { text },
  attribute src { text },
  [ a:defaultValue = "no" ] attribute loop { Yes_No }?

line_breaking_method =
  element attlist_line_breaking_method {
    attlist_kinsoku,
    top_prohibit_char?,
    end_prohibit_char?,
    hanging_char?
  }
attlist_line_breaking_method &=
  attribute method { "none" | "run_down" }?,
  attribute hanging_punctuation { Yes_No }?
kinsoku =
  element kinsoku {
    attlist_kinsoku,
    top_prohibit_char?,
    end_prohibit_char?,
    hanging_char?
  }
attlist_kinsoku &=
  attribute method { "none" | "run_down" }?,
  attribute hanging_punctuation { Yes_No }?
top_prohibit_char =
  element top_prohibit_char { attlist_top_prohibit_char, text }
attlist_top_prohibit_char &= empty
end_prohibit_char =
  element end_prohibit_char { attlist_end_prohibit_char, text }
attlist_end_prohibit_char &= empty
hanging_char = element hanging_char { attlist_hanging_char, text }
attlist_hanging_char &= empty
Dict_Block = p | ( scrolling_text | telop )
Dict_Inline =
  br | hr | font | ( horizontal | yoko ) | ruby | external_char | mask | char_id
Dict_Tag = text | Dict_Block | Dict_Inline | object
dict_body = element dict_body { attlist_dict_body, word+ }
```

```
attlist_dict_body &= empty
word = element word { attlist_word, (Dict_Tag | key_entry)* }
attlist_word &=
  attribute page_break { Yes_No }?,
  attribute turning_page_control { Turn_Page_Val }?,
  attribute head_button_control { Turn_Page_Val }?,
  attribute revision { text }?,
  attribute delete { Yes_No }?
key_entry = element key_entry { attlist_key_entry, key_item+ }
attlist_key_entry &= empty
key_item = element key_item { attlist_key_item, TextWithGaiji }
attlist_key_item &=
  attribute search_word { text },
  attribute table_id { text }
p = element p { attlist_p, Dict_Tag* }
attlist_p &=
  attribute top_line_indent { text }?,
  attribute top { text }?,
  attribute bottom { text }?,
  attribute align { "center" | "right" | "left" }?
scrolling_text =
  element scrolling_text {
    attlist_scrolling_text, (text | external_char | font | ( horizontal | yoko ) | ruby | object)*
  }
attlist_scrolling_text &= empty
telop =
  element telop {
    attlist_telop, (text | external_char | font | ( horizontal | yoko ) | ruby | object)*
  }
attlist_telop &= empty
br = element br { attlist_br, empty }
attlist_br &= attribute clear { "left" | "right" | "all" }?
```

```
hr = element hr { attlist_hr, empty }
attlist_hr &=
  attribute size { text }?,
  attribute length { text }?,
  attribute align { "left" | "center" | "right" }?
font = element font { attlist_font, (text | Dict_Inline | object)* }
attlist_font &=
  attribute name { text }?,
  attribute size { text }?,
  attribute base { "default" | "last" }?,
  attribute color_space { "RGB" }?,
  attribute opacity { "100" }?,
  attribute color { text }?,
  attribute bold { Yes_No }?,
  attribute underline { Yes_No }?
horizontal = element horizontal { attlist_horizontal, TextWithGaiji }
attlist_horizontal &= empty
yoko = element yoko { attlist_yoko, TextWithGaiji }
attlist_yoko &= empty
ruby = element ruby { attlist_ruby, rbase, rtop }
attlist_ruby &= empty
rbase = element rbase { attlist_rbase, TextWithGaiji }
attlist_rbase &= empty
rtop = element rtop { attlist_rtop, TextWithGaiji }
attlist_rtop &= empty
external_char = element external_char { attlist_external_char, text }
attlist_external_char &=
  attribute alt_set { text }?,
  attribute alt_code { text }?,
  attribute alt_img { text }?,
  attribute alt_vimg { text }?,
  attribute img_type { text }?,
  attribute alt { text }?
mask =
  element mask {
    attlist_mask,
    (text | br | font | ( horizontal | yoko ) | ruby | external_char | object)*
  }
attlist_mask &=
  [ a:defaultValue = "on" ] attribute initial_flag { "on" | "off" }?,
  attribute trigger { text }?,
  attribute char_id { text }?,
  [ a:defaultValue = "default" ]
  attribute mask_type { "default" | "color" }?,
  attribute color_space { "RGB" }?,
  attribute opacity { "100" }?,
  [ a:defaultValue = "black" ] attribute color { text }?,
  [ a:defaultValue = "scope" ] attribute hold_flag { HoldFlag }?
char_id =
  element char_id { attlist_char_id, (text | Dict_Inline | object)* }
attlist_char_id &= attribute char_id { text }
object = element object { attlist_object, permission_info? }
attlist_object &=
  attribute type { text },
  attribute src { text },
  attribute char_id { text }?,
  attribute align { "top" | "middle" | "bottom" | "left" | "right" }?,
  attribute start { "auto" | "event" }?,
  attribute loop { "1" }?
start = dict_data
```

B.7.4 Tags and attributes for text object instance 2

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

Yes_No = "yes" | "no"
TextWithGaiji = (text | external_char)*
```

```
BaseLine = "right" | "right_only" | "down" | "down_only"
HoldFlag = "scope" | "on_power" | "save"
Turn_Page_Val = "on" | "off" | "forward" | "back"
Paragraph_Align =
  "left"
  | "center"
  | "right"
  | "justified"
  | "justified_left"
  | "justified_center"
  | "justified_right"
permission_info =
  element permission_info {
    attlist_permission_info, print_permission?, copy_permission?
  }
attlist_permission_info &= empty
print_permission =
  element print_permission { attlist_print_permission, empty }
attlist_print_permission &=
  [ a:defaultValue = "no" ]
  attribute permission { "authorized" | "no" }?
copy_permission =
  element copy_permission { attlist_copy_permission, empty }
attlist_copy_permission &=
  [ a:defaultValue = "no" ]
  attribute permission { "authorized" | "in_device_only" | "no" }?
vertex = element vertex { attlist_vertex, empty }
attlist_vertex &= attribute position { text }
text_data2 =
  element text_data2 {
    attlist_text_data2, text_default_attribute?, text_body2
  }
```

```
attlist_text_data2 &= empty
text_default_attribute =
  element text_default_attribute {
    attlist_text_default_attribute,
    text_default_font?,
    text_default_background?,
    text_default_background_music?,
    ( kinsoku? | line_breaking_method? )
  }
attlist_text_default_attribute &=
  attribute baseline { BaseLine }?,
  attribute valign { "middle" }?
text_default_font =
  element text_default_font { attlist_text_default_font, empty }
attlist_text_default_font &=
  attribute fontname { text }?,
  attribute color_space { "RGB" }?,
  attribute opacity { "100" }?,
  attribute color { text }?,
  attribute ruby_flag { "yes" | "yes_only" | "no" | "no_only" }?
text_default_background =
  element text_default_background {
    attlist_text_default_background, permission_info?
  }
attlist_text_default_background &=
  attribute color_space { "RGB" }?,
  attribute opacity { "100" }?,
  attribute color { text }?,
  attribute type { text }?,
  attribute src { text }?

text_default_background_music =
  element text_default_background_music {
    attlist_text_default_background_music, permission_info?
  }
attlist_text_default_background_music &=
  attribute type { text },
  attribute src { text },
  [ a:defaultValue = "no" ] attribute loop { Yes_No }?

line_breaking_method =
  element line_breaking_method {
    attlist_line_breaking_method,
    top_prohibit_char?,
    end_prohibit_char?,
    hanging_char?
  }
attlist_line_breaking_method &=
  attribute method { "none" | "run_down" }?,
  attribute hanging_punctuation { Yes_No }?
kinsoku =
  element kinsoku {
    attlist_kinsoku,
    top_prohibit_char?,
    end_prohibit_char?,
    hanging_char?
  }
attlist_kinsoku &=
  attribute method { "none" | "run_down" }?,
  attribute hanging_punctuation { Yes_No }?
top_prohibit_char =
  element top_prohibit_char { attlist_top_prohibit_char, text }
attlist_top_prohibit_char &= empty
end_prohibit_char =
  element end_prohibit_char { attlist_end_prohibit_char, text }
attlist_end_prohibit_char &= empty
hanging_char = element hanging_char { attlist_hanging_char, text }
attlist_hanging_char &= empty
Text2_Block = p | ( scrolling_text | telop ) | key_entry
Text2_Inline =
  br
  | hr
  | font
  | ( horizontal | yoko )
  | ruby
  | external_char
  | mask
  | char_id
  | meaning_start
  | select
Text2_All_Tag2 = text | Text2_Block | Text2_Inline | object
Text2_All_Tag = Text2_All_Tag2 | page_break
text_body2 = element text_body2 { attlist_text_body2, Text2_All_Tag* }
attlist_text_body2 &= empty
p = element p { attlist_p, Text2_All_Tag2* }
attlist_p &=
  attribute top_line_indent { text }?,
  attribute top { text }?,
  attribute bottom { text }?,
  attribute align { Paragraph_Align }?
scrolling_text =
  element scrolling_text {
    attlist_scrolling_text,
    (text | external_char | font | ( horizontal | yoko ) | ruby | object | select)*
  }
attlist_scrolling_text &= empty
```

```
telop =
  element telop {
    attlist_telop,
    (text | external_char | font | ( horizontal | yoko ) | ruby | object | select)*
  }
attlist_telop &= empty
page_break = element page_break { attlist_page_break, empty }
attlist_page_break &=
  attribute turning_page_control { Turn_Page_Val }?,
  attribute head_button_control { Turn_Page_Val }?
br = element br { attlist_br, empty }
attlist_br &=
  attribute clear { "left" | "right" | "all" }?,
  attribute width { text }?
hr = element hr { attlist_hr, permission_info? }
attlist_hr &=
  attribute size { text }?,
  attribute length { text }?,
  attribute align { "left" | "center" | "right" }?,
  attribute type { text }?,
  attribute h_src { text }?,
  attribute v_src { text }?

font = element font { attlist_font, (text | Text2_Inline | object)* }
attlist_font &=
  attribute name { text }?,
  attribute size { text }?,
  attribute base { "default" | "last" }?,
  attribute color_space { "RGB" }?,
  attribute opacity { "100" }?,
  attribute color { text }?,
  attribute bold { Yes_No }?,
  attribute underline { Yes_No }?
horizontal = element horizontal { attlist_horizontal, TextWithGaiji }
attlist_horizontal &= empty
yoko = element yoko { attlist_yoko, TextWithGaiji }
attlist_yoko &= empty
ruby = element ruby { attlist_ruby, rbase, rtop }
attlist_ruby &= empty
rbase = element rbase { attlist_rbase, TextWithGaiji }
attlist_rbase &= empty
rtop = element rtop { attlist_rtop, TextWithGaiji }
attlist_rtop &= empty
external_char = element external_char { attlist_external_char, text }
attlist_external_char &=
  attribute alt_set { text }?,
  attribute alt_code { text }?,
  attribute alt_img { text }?,
  attribute alt_vimg { text }?,
  attribute img_type { text }?,
  attribute alt { text }?
mask =
  element mask {
    attlist_mask,
    (text | br | font | yoko | ruby | external_char | object | select)*
  }
attlist_mask &=
  [ a:defaultValue = "on" ] attribute initial_flag { "on" | "off" }?,
  attribute trigger { text }?,
  attribute char_id { text }?,
  [ a:defaultValue = "default" ]
  attribute mask_type { "default" | "color" }?,
  attribute color_space { "RGB" }?,
  attribute opacity { "100" }?,
  [ a:defaultValue = "black" ] attribute color { text }?,
  [ a:defaultValue = "scope" ] attribute hold_flag { HoldFlag }?
char_id =
  element char_id { attlist_char_id, (text | Text2_Inline | object)* }
attlist_char_id &= attribute char_id { text }
select = element select { attlist_select, select_item+ }
attlist_select &=
  [ a:defaultValue = "baseline" ] attribute variable { "baseline" }?
select_item =
  element select_item {
    attlist_select_item,
    (text | font | yoko | ruby | external_char | object)*
  }
attlist_select_item &= attribute type { "right" | "down" }
object = element object { attlist_object, permission_info? }
attlist_object &=
  attribute type { text },
  attribute src { text },
  attribute char_id { text }?,
  attribute align { "top" | "middle" | "bottom" | "left" | "right" }?,
  attribute start { "auto" | "event" }?,
  attribute loop { "1" }?,
  attribute position { "on_edge" }?

key_entry = element key_entry { attlist_key_entry, key_item+ }
attlist_key_entry &= empty
key_item = element key_item { attlist_key_item, TextWithGaiji }
attlist_key_item &=
  attribute search_word { text },
  attribute table_id { text }
meaning_start = element meaning_start { attlist_meaning_start, empty }
attlist_meaning_start &= empty
```

```
Area_Tag = dynamic_text_area | object_area
layout_info_list =
  element layout_info_list { attlist_layout_info_list, layout_info+ }
attlist_layout_info_list &= empty
layout_info =
  element layout_info {
    attlist_layout_info,
    outline_info?,
    layout_default_attribute?,
    Area_Tag*
  }
attlist_layout_info &=
  attribute baseline { "right" | "down" }?,
  attribute display_size { text }
outline_info =
  element outline_info {
    attlist_outline_info,
    outline_polygon?,
    outline_background?,
    outline_background_music?
  }
attlist_outline_info &= attribute rect { text }?
outline_polygon =
  element outline_polygon { attlist_outline_polygon, vertex* }
attlist_outline_polygon &=
  attribute line_color { text }?,
  attribute line_width { text }?
outline_background =
  element outline_background {
    attlist_outline_background, permission_info?
  }
attlist_outline_background &=
  attribute src { text }?,
  attribute type { text }?,
  attribute color_space { "RGB" }?,
  attribute opacity { "100" }?,
  attribute color { text }?

outline_background_music =
  element outline_background_music {
    attlist_outline_background_music, permission_info?
  }
attlist_outline_background_music &=
  attribute src { text },
  attribute type { text },
  [ a:defaultValue = "no" ] attribute loop { Yes_No }?

dynamic_text_area =
  element dynamic_text_area { attlist_dynamic_text_area, empty }
attlist_dynamic_text_area &=
  attribute rect { text },
  attribute bg_color { text }?,
  attribute bg_color_space { "RGB" }?,
  attribute bg_opacity { "100" }?
object_area =
  element object_area { attlist_object_area, permission_info? }
attlist_object_area &=
  attribute rect { text },
  attribute src { text },
  attribute type { text },
  attribute char_id { text }?,
  attribute bg_color { text }?,
  attribute bg_color_space { "RGB" }?,
  attribute bg_opacity { "100" }?
```

```
layout_default_attribute =
  element layout_default_attribute {
    attlist_layout_default_attribute,
    layout_line_breaking_method?,
    layout_default_size?,
    layout_default_paragraph?,
    layout_default_font?,
    layout_default_columns?
  }
attlist_layout_default_attribute &=
  attribute valign { "middle" | "end" }?
layout_line_breaking_method =
  element layout_line_breaking_method {
    attlist_layout_line_breaking_method,
    top_prohibit_char?,
    end_prohibit_char?,
    hanging_char?
  }
attlist_layout_line_breaking_method &=
  attribute method { "none" | "run_down" }?,
  attribute hanging_punctuation { Yes_No }?
layout_default_size =
  element layout_default_size { attlist_layout_default_size, empty }
attlist_layout_default_size &=
  attribute letter_spacing { text }?,
  attribute line_pitch { text }?,
  attribute margin { text }?
layout_default_paragraph =
  element layout_default_paragraph {
    attlist_layout_default_paragraph, empty
  }
```

```
attlist_layout_default_paragraph &=
  attribute top_line_indent { text }?,
  attribute top { text }?,
  attribute bottom { text }?,
  attribute align { Paragraph_Align }?
layout_default_font =
  element layout_default_font { attlist_layout_default_font, empty }
attlist_layout_default_font &=
  attribute fontname { text }?,
  [ a:defaultValue = "yes" ] attribute fname_change_flag { "yes" }?,
  attribute fontsize { text }?,
  attribute rubysize { text }?,
  attribute ruby_flag { "yes" | "yes_only" | "no" | "no_only" }?,
  attribute bold_flag { Yes_No }?,
  [ a:defaultValue = "off" ]
  attribute fsize_autochange_flag { "on" | "off" }?,
  attribute color_space { "RGB" }?,
  attribute opacity { "100" }?,
  attribute color { text }?
layout_default_columns =
  element layout_default_columns {
    attlist_layout_default_columns, columns_ruled_line?
  }
attlist_layout_default_columns &=
  attribute line_character_max_count { text }?,
  attribute line_character_min_count { text }?,
  attribute tray_spacing { text }?
columns_ruled_line =
  element columns_ruled_line {
    attlist_columns_ruled_line, permission_info?
  }
attlist_columns_ruled_line &=
  attribute color_space { "RGB" }?,
  attribute opacity { "100" }?,
  attribute color { text }?,
  attribute width { text }?,
  attribute h_src { text }?,
  attribute v_src { text }?,
  attribute type { text }?,
  attribute length { text }?,
  attribute align { "top" | "center" | "bottom" }?
start = text_data2 | layout_info_list
```

Annex C
(normative)

ESP format

C.1 General

This annex describes ESP format. mentioned in Clause 6.

ESP format is a data interchange format for electronic books with the goal of making the long-term reuse of content possible without being constrained to a target terminal while including the functionality of e-book content that has been accumulated.

ESP format is primarily envisioned to function as an intermediate format before e-books are deployed to various terminals and platforms.

In developing the specifications, it was regarded important to separate content and style to enable long-term reuse of e-book data while maintaining sufficient capability to describe the existing e-books.

This standard uses XHTML/CSS notations (names of tags/attributes/properties/values and their usages) where appropriate.

C.2 Overview

C.2.1 Content structure

The data that makes up ESP format is broadly divided into the categories in Table C.1.

The content in ESP format is described in the XML format. To specify how that content is displayed, a file shall be prepared that describes the style data.

For each type of data, the data is stored in a separate file that uses the root element defined in Table C.1. Therefore, a minimum of four files (when the content does not possess style data) or five files (when the content possesses style data) are included in a single item of content.

The file name of the XML file that indicates the table of contents/item data is fixed as package.xml.

In addition to the data listed above, if there is data such as images, audio, or video that is specified from the data, this data shall also be included in the content.

For details on notation for the data, see Clause C.3 and Clause C.4.

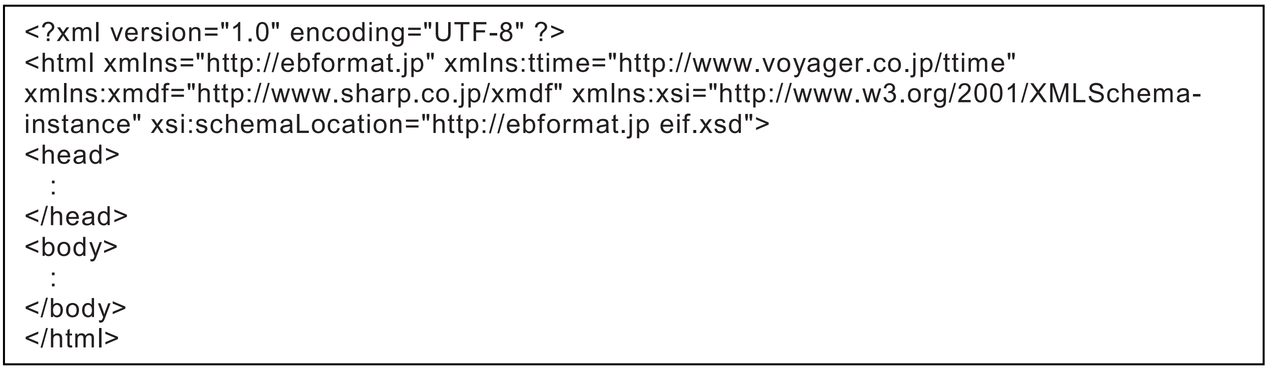
Schema files are means for representing the format specifications and do not constitute the specifications (the file names shown in the table below are just examples and it suffices that the schema file names are consistent with those indicated in XML files).

Table C.1 – ESP format structure

Data type	Schema file name	Extension	Root element name	Notes	Required
Body data	eif.xsd xml.xsd	xml	html	Includes tags related to text, dictionaries, and comics. The body data for the content may be split into multiple files.	✓
Bibliography data	eif.xsd xml.xsd	xml	bibliography	Data related to the bibliography including the title and author.	✓
Global settings data	eif.xsd xml.xsd	xml	global_setting	Configures the contents overall. Primarily configures settings related to the viewer not included in the style.	
Table of contents/item data	eif.xsd xml.xsd	xml	package	File name is fixed as package.xml.	✓
Search table	eif.xsd xml.xsd	xml	search_table	Tags that describe the search table. This file shall be included when describing dictionary content.	
Style data		css		Corresponds to the style sheet. When anticipating multiple configurations, store each of those in a separate file.	

C.2.2 XML files

An example of content XML file is shown in Figure C.1.



```
<?xml version="1.0" encoding="UTF-8" ?>
<html xmlns="http://ebformat.jp" xmlns:ttime="http://www.voyager.co.jp/ttime"
xmlns:xmdf="http://www.sharp.co.jp/xmdf" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://ebformat.jp eif.xsd">
<head>
   :
</head>
<body>
   :
</body>
</html>
```

IEC 2806/13

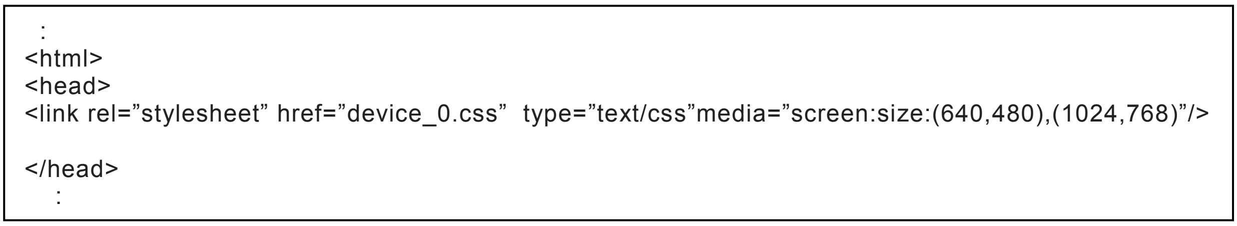
Figure C.1 – Content XML file example

C.2.3 The relationship between body data and style data

In ESP format, the display method (style) is not defined as a part of the tag specification. This style is represented by the style data.

To handle body data with style data, code it with the <link> tag, a child element of the <head> tag that is always included in the body data XML. An example is shown in Figure C.2 For details, see Clause C.3.

For the properties used with style data, see C.4.1. For the selectors that can be used for each property, see C.4.3.



```
  :
<html>
<head>
<link rel=”stylesheet” href=”device_0.css”  type=”text/css”media=”screen:size:(640,480),(1024,768)”/>

</head>
    :
```

IEC 2807/13

Figure C.2 – Style data example

C.2.4 Character codes

Character codes are written according to normal XML notation. However, for characters that cannot be expressed in this manner, a coding method has also been prepared. (See Clause C.3 for details.)

C.2.5 Line break codes

The line break codes shown in Table C.2 are all usable. See C.2.7.

Table C.2 – Usable line break codes

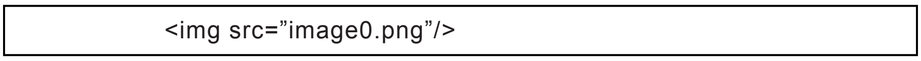
Line break	ASCII code
CR + LF	0x0d + 0x0a (2 bytes)
CR	0x0d
LF	0x0a

C.2.6 Coding precautions

C.2.6.1 Precautions related to the XML specification

Since ESP format is coded in XML, take note of the following items.

- There is a distinction between uppercase characters and lowercase characters in XML tag names, so do not use uppercase characters in tags unless specified by the specification.
- Following the XML specification, the end tag shall not be omitted. For an empty tag with no end tag, always end the tag with "/>". (See Figure C.3).
- Attributes shall have attribute values enclosed in single quotes (0x27) or double quotes (0x22). (See Figure C.3).
- The separation between different attributes is either a half-width space (0x20), a tab (0x09), a CR (0x0d), or a LF (0x0a). The number of separating characters does not affect parsing.
- The order of attributes within the same tag does not affect parsing.



```
<img src=”image0.png”/>
```

IEC 2808/13

Figure C.3 – Tag coding example

C.2.6.2 Precautions related to the CSS specification

- To express length or size, units may be added to numeric values. When a value is not "0", the unit shall be added.

C.2.6.3 Other precautions

- This standard does not stipulate a folder structure, but for reuse in different environments, individual file names shall be coded as relative paths.
- Use "/" (0x2f), not "\" (0x5c) for the path name separator.
- Excluding the path name separator, the characters that can be used in file names are listed below.
  - Half-width lowercase alphabetic characters and half-width uppercase alphabetic characters (a to z, A to Z). However, do not give file names to different files that will be the same when lowercase characters and uppercase characters are handled as the same characters.
  - Half-width numerals (0 to 9)
  - Half-width symbols shown in Table C.3
- In general, attributes and style sheet properties not listed in C.3 and C.4 cannot be used even if they are allowed in XHTML.

Table C.3 – Half-width symbols for file names

Character	Character code	Notes
Hyphen	0x2d	
Period	0x2e	
Underscore	0x5f	
Tilde	0x7e	
Percent	0x25	Used only in URL encoding of characters that are not allowed to be used.
Plus	0x2b	Used only in URL encoding of characters that are not allowed to be used.

C.2.7 Restrictions

C.2.7.1 General

To prevent development of ESP-format-related tools from becoming complicated the following restrictions are recommended when making the content data. Note that capabilities of the format are not affected by doing so.

C.2.7.2 Line break codes

Only CR+LF or LF are used as the line break code. Specifically, a one-byte CR by itself is not used as a line break code.

C.2.7.3 Specificity and coding order

For style coding, the code with high specificity is put later.

Specificity is computed by class attribute designation * 10 + tag designation count.

Example: a {color:red;} has a specificity of 1
.class {color;red;} has a specificity of 10
a.class {color;red;} has a specificity of 10 + 1 = 11

Therefore, they shall be written in this order.

NOTE If a conflict occurs when the specificity is the same, the latter code is given precedence.

C.2.7.4 Shorthand codes

C.2.7.4.1 General

For style code, the types of shorthand codes shown below are to be avoided.

C.2.7.4.2 Shorthand notation

(a) Color codes

Not acceptable: color:#fff;

Acceptable: color:#ffffff;

(b) Background color

Not acceptable: background: #a0a0a0;

Acceptable: background-color: #a0a0a0;

C.2.7.4.3 Shorthand designations

Not acceptable: border:5px;

Acceptable: border-top:5px; border-right:5px; border-bottom:5px; border-left:5px;

Remark: This also applies to other properties such as background, margin, padding.

C.2.7.4.4 Grouped designations

Not acceptable: . class1, . class2 {color:red;background-color:silver;}

Acceptable: . class1 {color:red;background-color:silver;}

. class2 {color:red;background-color:silver;}

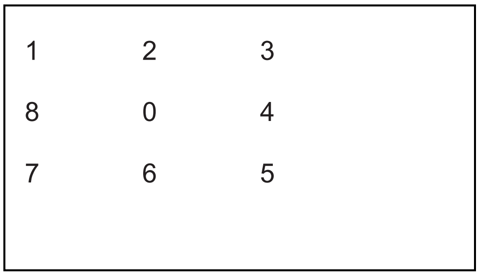
C.3 Elements and attributes

C.3.1 General

The notations used in this standard are explained in Table C.4.

Table C.4 – Notations

Notation	Meaning	Notes
URI	Indicates file names, file paths, and resources on a network.	
bool	Boolean value (true or false)	
colorvalue	Color designation	
[n]	Integer (an integer of 0 or 1 and higher)	
[i]	Integer (may be negative)	
posnumber	A number that indicates the origin point for the absolute position on the page (see Figure C.4)	For example, when 0, it indicates the origin point is the center of the page.



IEC 2809/13

Figure C.4 – Numbers that indicate the origin point for the absolute position on the page

Remarks

– Empty tags that do not have child elements have (empty tag) written after the tag name.

– The following terms are used to described parent-child relationship between elements.

block level elements	div, h1, h2, h3, h4, h5, h6, h7, h8, h9, marquee, table, action, tts, section, page_break, reghead, hr
inline elements	br, em, span, external_char, mlg, sub, sup, phead, ptail, map, ruby, case, nocase, mask, column_break, a, offset
object elements	img, audio, video

– Attributes that cannot be omitted are presented with “[required]”. Those that can be omitted are presented with “ [optional]”.

C.3.2 Common tags

<html>

The document's top level element (root tag).
Possible child elements: head, body
The <html> tag has the following attributes.

[Attributes]

xmlns [optional]
Defines the namespace used in the document.

xml:lang [optional]
Defines the language used in the document.

<head>

Document information. Child element of <html>.
Possible child elements: title, link, pdef, local_setting

<title>

Document title. Child element of <head>.
Possible child elements: text data, external_char, ruby

<body>

Document body. Child element of <html>.
Possible child elements: text data, block-level elements, inline elements, object elements, search_table, comic_cell_type_body, comic_page_type_body, dict_item, ref, split, column, search_page.

The <body> tag has the following attributes.

[Attributes]

type [optional]
Indicates the document type.
The default is "text".
This attribute takes the following value(s).
"text"
Text

"search"
Search table
"comic"
Comic
"dict"
Dictionary

turning_page_control [optional]
Specifies the behavior when turning the page.
This attribute takes the following value(s).
"on"
Neither the page for the previous flow nor the page for the next flow can be navigated to.
"off"
Both the page for the previous flow and the page for the next flow can be navigated to.
"forward"
The next flow cannot be navigated to. The previous flow can be navigated to.
"back"
The previous flow cannot be navigated to. The next flow can be navigated to.
"lastpage"
Indicates that this page is the last page.
This attribute value need not be added to the last page. This attribute value has been adopted for compatibility.

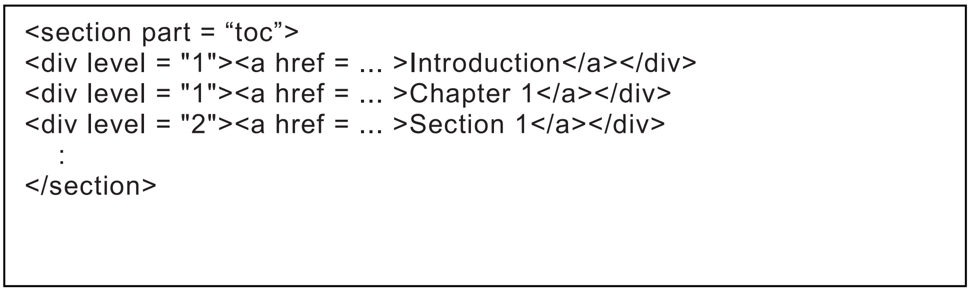
class [optional]
Gives a grouping to the element. Given in text data.

<section>
Indicates a section. In coding the <section> tag, if the part attribute is "toc" for example, note that this only logically shows that this part is the table of contents. Refer to the usage example for how to use this tag.
Possible child elements: text data, block-level elements, inline elements, object elements
The <section> tag has the following attributes.

[Attributes]

part [optional]
Indicates the purpose of this part.
This attribute takes the following value(s).
"toc"
Table of contents
"cover"
Front cover

The example of the usage of <section> is shown in Figure C.5.



```
<section part = “toc”>
<div level = "1"><a href = ... >Introduction</a></div>
<div level = "1"><a href = ... >Chapter 1</a></div>
<div level = "2"><a href = ... >Section 1</a></div>
   :
</section>
```

IEC 2810/13

Figure C.5 – <section> tag coding example

Remarks:

- The child elements <div> and <a> need not be included.
- The level attribute for <div> is envisioned as matching the heading level for the destination (if level="2", the destination is <H2>), but as a part of the format specification, when there is an <a> tag in a child element in the <section> tag, the

destination need not be <Hn>.

<local_setting>

A valid (local) setting in this entire file. Child element of <head>.
Possible child elements: default_ccs, page_progression_direction, window, proprietary

<permission>

Permission information for printing, copying, and other actions. To make the relevant files a subject for permissions, code as a child element of <item> in package.xml. To make the content overall a subject for permissions, code as a child element of <global_setting>.
Possible child elements: text data, block-level elements, inline elements, object elements
The <permission> tag has the following attributes.

[Attributes]

type [required]

Operations which are the subject of permission information
This attribute takes the following value(s).

"print"
Print

"copy"
Copy

"reading"
Reading setting

"export"
Export setting

"download"
Downloadable

"irda"
Permission for IR communication to Palm

"reading_on_ppc"
Permission to read on PocketPC

"conv_to_palm"
Permission to output for Palm via T-Break

value [optional]

The attribute related to copy permission
This attribute takes the following value(s).

"authorized"
Allow for only legitimate users.

"in_device_only"
For legitimate users, allow usage limited to within terminals where the bunko viewer is installed. So that content is not copied off the terminal, copying shall not function for terminals that cannot be managed.

"no"
Do not allow, even for legitimate users.

C.3.3 Text-related tags

<page_break> (empty tag)
Page break
Possible child elements: none
The <page_break> tag has the following attributes.

[Attributes]

method [optional]

Restricts page breaks. Specify collation for the next paginated page with this tag.
This attribute takes the following value(s).

"normal"
Normal

"odd"
Start from odd pages

"even"
Start from even pages

"left"
Start from the left page regardless of the page turning direction

"right"

Start from the right page regardless of the page turning direction

turning_page_control [optional]

Specifies the behavior when turning the page.

This attribute takes the following value(s).

"on"

Neither the data before this tag nor after this tag can be navigated to.

"off"

Both the data before this tag and after this tag can be navigated to.

"forward"

The data after this tag cannot be navigated to. The data before this tag can be navigated to.

"back"

The data before this tag cannot be navigated to. The data after this tag can be navigated to.

"lastpage"

The last page.

<column_break> (empty tag)

Column break

Possible child elements: none

<br> (empty tag)

Line break

Possible child elements: none

The <br> tag has the following attributes.

[Attributes]

clear [optional]

Specifies clearing text that wraps around images or other elements. Displays the next line underneath.

Remarks: Cannot be used with strict, but the clear attribute was used because the CSS clear property can only be used for block elements.

This attribute takes the following value(s).

"all"

Inserts a blank line up to the location where text can be displayed. For horizontal text, up to the left and right edges of the line. For vertical text, up to the bottom and top edges of the line.

"left"

Inserts a blank line up to the location where text can be displayed from the left edge of the line (for horizontal text. Top edge of the line for vertical text).

"right"

Inserts a blank line up to the location where text can be displayed from the right edge of the line (for horizontal text. Bottom edge of the line for vertical text).

class [optional]

Gives a grouping to the element. Given in text data.

<div>

General purpose block level element.

Possible child elements: text data, block-level elements, inline elements, object elements

The <div> tag has the following attributes.

[Attributes]

level [optional]

Block level

This attribute takes the following value(s).

Numeric value

id [optional]

Gives an identifier to the element. Must be unique in the XML file. Given in text data.

class [optional]

Gives a grouping to the element. Given in text data.

<em>
Emphasis.
Possible child elements: text data, inline elements, object elements, a, offset
The <em> tag has the following attributes.
class [optional]
Gives a grouping to the element. Given in text data.

<h1>
Heading (level 1)
Possible child elements: text data, inline elements, object elements, div
The <h1> tag has the following attributes.
[Attributes]
caption [optional]
Text to display in the running head
This attribute takes the following value(s).
Text
class [optional]
Gives a grouping to the element. Given in text data.

<h2>
Heading (level 2)
Possible child elements: text data, inline elements, object elements, div
The <h2> tag has the following attributes.
[Attributes]
caption [optional]
Text to display in the running head
This attribute takes the following value(s).
Text
class [optional]
Gives a grouping to the element. Given in text data.

<h3>
Heading (level 3)
Possible child elements: text data, inline elements, object elements, div
The <h3> tag has the following attributes.
[Attributes]
caption [optional]
Text to display in the running head
This attribute takes the following value(s).
Text
class [optional]
Gives a grouping to the element. Given in text data

<h4>
Heading (level 4)
Possible child elements: text data, inline elements, object elements, div
The <h4> tag has the following attributes.
[Attributes]
caption [optional]
Text to display in the running head
This attribute takes the following value(s).
Text
class [optional]
Gives a grouping to the element. Given in text data.

<h5>
Heading (level 5)
Possible child elements: text data, inline elements, object elements, div
The <h5> tag has the following attributes.
[Attributes]
caption [optional]
Text to display in the running head

This attribute takes the following value(s).
Text
class [optional]
Gives a grouping to the element. Given in text data.

<h6>
Heading (level 6)
Possible child elements: text data, inline elements, object elements, div
The <h6> tag has the following attributes.
[Attributes]
caption [optional]
Text to display in the running head
This attribute takes the following value(s).
Text
class [optional]
Gives a grouping to the element. Given in text data.

<h7>
Heading (level 7)
Possible child elements: text data, inline elements, object elements, div
The <h7> tag has the following attributes.
[Attributes]
caption [optional]
Text to display in the running head
This attribute takes the following value(s).
Text
class [optional]
Gives a grouping to the element. Given in text data.

<h8>
Heading (level 8)
Possible child elements: text data, inline elements, object elements, div
The <h8> tag has the following attributes.
[Attributes]
caption [optional]
Text to display in the running head
This attribute takes the following value(s).
Text
class [optional]
Gives a grouping to the element. Given in text data.

<h9>
Possible child elements: text data, inline elements, object elements, div
Heading (level 9)
The <h9> tag has the following attributes.
[Attributes]
caption [optional]
Text to display in the running head
This attribute takes the following value(s).
Text
class [optional]
Gives a grouping to the element. Given in text data.

<span>
General purpose inline element.
Possible child elements: text data, inline elements, object elements
Remarks: To write alternate text, it can also be written as a child element of <external_char> (see the remark for the <external_char> tag).
The <span> tag has the following attributes.
[Attributes]
id [optional]

Gives an identifier to the element. Must be unique in the XML file. Given in text data.

type [optional]

When a reference to a media file is enclosed, specify the media file type in the MIME format.

xml:lang [optional]

Specifies the language.

lang_code [optional]

Specifies the language code.
This attribute takes the following value(s).
Language code

class [optional]

Gives a grouping to the element. Given in text data.

<offset> (empty tag)

Offset of the display position. Use in combination with the style data property text-offset.
Possible child elements: none

class [optional]

Gives a grouping to the element. Given in text data.

<external_char>

Inserts external characters.
Possible child elements: span,img
The <external_char> tag has the following attributes.

[Attributes]

system [optional]

Specifies the presumed code system. Example: "Adobe-Japan-1-6"

code [optional]

The code in the code system specified by system

glyph_data [optional]

Specifies the external character image (file containing the glyph).

priority [optional]

Explicitly expresses whether the code or glyph_data attribute takes priority when conflicts occur. If code takes priority over glyph, then “code,glyph”. And otherwise “glyph,code”, the two words being separated by a comma.”

alt [optional]

Alternate text
Remarks: Alternate text can also be written using the <span> tag or the <img> tag in a child element.

Remarks: To specify alternate text and alternate images, you may use the <span> tag and the <img> tag as shown below in addition to using the alt attribute and the glyph_data attribute.
Example 1: An example using <span>

```
<external_char glyph_data="img/glyph.png">
  <span class=”yoko”>!?</span>
</external_char>
...
span.yoko { text-combine: horizontal; }
```

Example 2: An example using <img>

```
<external_char glyph_data="img/glyph.png">
  <img src=”img/alt_glyph.jpg”/>
</external_char>
```

- Of the attributes specified in <img>, a child element of <external_char>, attributes other than the src attribute shall not be written.

<mlg>

Insert note
Possible child elements: text data, inline elements, object elements

<a>
Anchor (link). The details of the operation are coded with either the href, target, ref_id/ref_itemid, or cmd attributes.
Possible child elements: text data, inline elements (except <a>), object elements
The <a> tag has the following attributes.

[Attributes]

href [optional]
Specifies a link.

target [optional]
Specifies the target.

id [optional]
Gives an identifier to the element. Must be unique in the XML file. Given in text data.

ref_itemid [optional]
Specifies the id of a referenced file (text or media file). If the reference is a media file, that media file starts playing.

title [optional]
Specifies the text to display in the title bar on rollover.

ref_id [optional]
Specifies the id of the reference.
Remarks: Independently added: source id

shake [optional]
Specifies whether or not to vibrate when clicked.
This attribute takes the following value(s).
bool

background [optional]
Whether or not to execute in the background.
This attribute takes the following value(s).
bool

cmd [optional]
Specifies the protocol to execute.
This attribute takes the following value(s).
"appl:"creator
Creator of the application to open
"exe:"exename
Process name of the application to open
"tel:"telephone number
Places a telephone call.
"mailto:"email address
Starts the mailer.
"nexthead"
To the next chapter
"prevhead"
To the previous chapter
"next"
To the next page
"prev"
To the previous page
"top"
To the top
"bottom"
To the end
"back"
Back
"forward"
Forward (in history)

class [optional]
Gives a grouping to the element. Given in text data.

Remarks:

1) href
2) ref_id / ref_itemid
3) cmd

For the attributes in 1) through 3) above, they can be combined under the following conditions, and the parsing of the attributes in those situations is written below.

・1 & 3 and 2 & 3 may be written in the same tag, but 1 & 2 shall not be written in the same tag.

If they are written in the same tag, 2 has precedence.

・If 1 & 3 conflict or 2 & 3 conflict, 3 has precedence.

(Conflict example: Jumps to different pages are specified by 2 & 3.)

・The target attribute can only be used with 1 & 2, and it cannot be used with 3.

<hr> (empty tag)
Displays a horizontal line.
Possible child elements: none
The <hr> tag has the following attributes.

class [optional]
Gives a grouping to the element. Given in text data.

<sub>
Displays text as a subscript.
Possible child elements: text data, inline elements, object elements

<sup>
Displays text as a superscript.
Possible child elements: text data, inline elements, object elements

<marquee>
The text enclosed by this tag is displayed in a scrolling ticker (made to look as if it is flowing).
Possible child elements: text data, external_char, ruby, object elements, span

<table>
Creates a table.
Possible child elements: tr

<th>
Adds a heading to the table. Child element of <tr>.
Possible child elements: text data, inline elements, object elements
The <th> tag has the following attributes.

[Attributes]

rowspan [optional]
Specifies the number of cells to span downward.
This attribute takes the following value(s).

[n]
0 or larger integer. Specify 0 to span all the cells downward including the attribute's own cell.

colspan [optional]
Specifies the number of cells to span to the right.
This attribute takes the following value(s).

[n]
0 or larger integer. Specify 0 to span all the cells to the right including the attribute's own cell.

<tr>
Defines a horizontal row. Child element of <table>.
Possible child elements: th, td

<td>
Creates a cell. Child element of <tr>.

Possible child elements: text data, inline elements, object elements
The <td> tag has the following attributes.

[Attributes]

rowspan [optional]
Specifies the number of cells to span downward.
This attribute takes the following value(s).
[n]
0 or larger integer. Specify 0 to span all the cells downward including the attribute's own cell.

colspan [optional]
Specifies the number of cells to span to the right.
This attribute takes the following value(s).
[n]
0 or larger integer. Specify 0 to span all the cells to the right including the attribute's own cell.

<img> (empty tag)
Specifies an image. Or specifies a fill color.
Possible child elements: none
Remarks: Some IMG attributes are overwritten by CSS, but other than align, it is a bit strange to make them all CSS, so they have been made attributes. Alternate text may be written as a child element of <external_char> (see the remark for the <external_char> tag).
The <img> tag has the following attributes.

[Attributes]

src [required]
Specifies the image URI or to fill.
This attribute takes the following value(s).
URI
Specify the URI when displaying an image.
"paint:"
Specify when filling.

alt [optional]
Specifies alternate text.

longdesc [optional]
Specifies a reference for a long description about the image.

width [optional]
Specifies the width of the image to display
This attribute takes the following value(s).
[n]px, [n]%

height [optional]
Specifies the height of the image to display.
This attribute takes the following value(s).
[n]px, [n]%

usemap [optional]
Specifies the URI of an image-map to associate with the image.

line [optional]
Specifies the image size as text size (integer value 1 or higher).

scale [optional]
Specifies the image size scaled from the original size.
This attribute takes the following value(s).
[n]%

bordercolor [optional]
Border color
This attribute takes the following value(s).
colorvalue

border [optional]
Border width
This attribute takes the following value(s).
[n]

hspace [optional]
Specifies the horizontal margins.

This attribute takes the following value(s).
[n]
0 or larger integer value.

vspace [optional]
Specifies the vertical margins.
This attribute takes the following value(s).
[n]
0 or larger integer value

dspace [optional]
Specifies the margins for the progression direction as an integer value 0 or higher.
This attribute takes the following value(s).
[n]
0 or larger integer value

snap [optional]
Snap to field frame
This attribute takes the following value(s).
"frame"
Snap to field frame

linemode [optional]
Specifies the inline image wrapping mode.
This attribute takes the following value(s).
bool

targetplane [optional]
Draws the image on the text buffer.
This attribute takes the following value(s).
"text"
Draws the image on the text buffer.

drawing [optional]
Tiling display
This attribute takes the following value(s).
"tile"
Tiling display

color [optional]
Specifies the fill color.
This attribute takes the following value(s).
colorvalue

opacity [optional]
Specifies the fill color opacity.
This attribute takes the following value(s).
[n]
Integer value from 0 to 255.

loop [optional]
Number of times to loop the animation
This attribute takes the following value(s).
[n]
1 or larger integer value.

start [optional]
Specifies the trigger to indicate the start of playback for an animation object when an animation object is specified by the src attribute.
This attribute takes the following value(s).
"auto"
Automatically starts playback from the beginning with each entry into the display region. Even if playback ended without completing the last time the animation object was displayed, playback starts from the beginning.
"event"
Starts playback as instructed by event data.

a [optional]
Sets the origin point when specifying the display position for the image.
This attribute takes the following value(s).
posNumber

x [optional]

The horizontal offset value from the origin point specified by a.
This attribute takes the following value(s).
[n]

y [optional]
The vertical offset value from the origin point specified by a.
This attribute takes the following value(s).
[n]

o [optional]
Specifies the image wrapping mode.
This attribute takes the following value(s).
bool

mode [optional]
Specifies the button display mode when used in combination with a tool button.
This attribute takes the following value(s).
[n]

shrink [optional]
Specifies the display method when the image size extends out from the page.
This attribute takes the following value(s).
screen
Shrinks the image so it is displayed in its entirety.

startpoint [optional]
Sets the origin point when enlarging a shrunken image.
This attribute takes the following value(s).
posNumber

preview [optional]
Controls the full scale display of the image. Enables full scale display mode.
This attribute takes the following value(s).
bool

id [optional]
Gives an identifier to the element. Must be unique in the XML file. Given in text data.

class [optional]
Gives a grouping to the element. Given in text data.

<pdef> (empty tag)
Illustration insertion tag. Specifies an image or a filling as the fixed visual objects in the page.
Possible child elements: none
Remarks: To be used in <head>...</head>
The <pdef> tag has the following attributes.

[Attributes]

src [required]
Specifies the image to set as the illustration or filling.
This attribute takes the following value(s).
URI
Specifies the URI of the image to set as the illustration.
"paint:"
Specifies filling the illustration.

id [required]
Gives an identifier to the image. Given in text data.

a [optional]
Specifies the origin point when specifying the display position of the image.
This attribute takes the following value(s).
posNumber

alt [optional]
Specifies alternate text.

width [optional]
Specifies the width of the image to display.
This attribute takes the following value(s).
[n]px, [n]%

height [optional]
Specifies the height of the image to display.

This attribute takes the following value(s).
[n]px, [n]%

usemap [optional]
Specifies the URI of an image-map to associate with the illustration.

bordercolor [optional]
Border color
This attribute takes the following value(s).
colorvalue

border [optional]
Border width
This attribute takes the following value(s).
[n]

hspace [optional]
Specifies the horizontal margins.
This attribute takes the following value(s).
[n]
0 or larger integer value.

vspace [optional]
Specifies the vertical margins.
This attribute takes the following value(s).
[n]
0 or larger integer value.

color [optional]
Specifies the fill color.
This attribute takes the following value(s).
colorvalue

opacity [optional]
Specifies the fill color opacity.
This attribute takes the following value(s).
[n]
Integer value from 0 to 255.

x [optional]
The horizontal offset value from the origin point specified by a.
This attribute takes the following value(s).
[n]

y [optional]
The vertical offset value from the origin point specified by a.
This attribute takes the following value(s).
[n]

o [optional]
Specifies the image wrapping mode.
This attribute takes the following value(s).
bool

mode [optional]
Button display when used in combination with a tool button
This attribute takes the following value(s).
[n]

shrink [optional]
Specifies the display method when the image size extends out from the page.
This attribute takes the following value(s).
screen
Shrinks the image so it is displayed in its entirety.

startpoint [optional]
Sets the origin point when enlarging a shrunken image.
This attribute takes the following value(s).
posNumber

preview [optional]
Controls the full scale display of the image. Enables full scale display mode.
This attribute takes the following value(s).
bool

<phead> (empty tag)

Begins to display the image specified by <pdef>. The image is shown until <ptail> is encountered in the page.
Possible child elements: none
The <phead> tag has the following attributes.

[Attributes]
ref_id [required]
Refers to one of the <pdef> tags to specify the image by the identifier. Must match the id attribute of one of the <pdef> tags in the content.
delay [optional]
Page offset value from the page with the tag.
This attribute takes the following value(s).
0 or larger integer value
Regarded as 0 if omitted.

<ptail> (empty tag)
Specifies the end point of display of the image specified by <pdef>
Possible child elements: none
The <ptail> tag has the following attributes.

[Attributes]
ref_id [required]
Refers to one of the <pdef> tags to specify the image by the identifier. Must match the id attribute of one of the <pdef> tags in the content.
delay [optional]
Page offset value from the page with the tag.
This attribute takes the following value(s).
0 or larger integer value
Regarded as 0 if omitted.

<area> (empty tag)
Specifies an image-map region.
Possible child elements: none
Remarks: The details of the operation are coded with either href, target, ref_id/ref_itemid, or type. For details, see the note for the <a> tag.
The <area> tag has the following attributes.

[Attributes]
alt [optional]
Text to display as an alternate
shape [optional]
Defines the shape of the region.
The default is "poly".
This attribute takes the following value(s).
"rect"
Rectangle
"circle"
Circle
"poly"
Polygon
"default"
Entire image
coords [optional]
Specifies the coordinates of the region. Specify in x coordinate, y coordinate order. Example: coords:"x,y"
href [optional]
Destination URI
target [optional]
Specifies the target (name of the frame to display in).
title [optional]
Specifies the text to display in the title bar on rollover.
ref_itemid [optional]
Specifies the id of a referenced file (text or media file). If the reference is a media file, that media file starts playing.
ref_id [optional]

Specifies the id of the reference.
Remarks: Independently added: source id

shake [optional]
Whether or not to vibrate when clicked.
This attribute takes the following value(s).
bool

background [optional]
Whether or not to execute in the background.
Remarks: (exclusive T-Time setting)
This attribute takes the following value(s).
bool

cmd [optional]
Specifies the protocol.
Remarks: Independently added: Place a telephone call, start mailer
This attribute takes the following value(s).

"appl:"creator
Creator of the application to open

"exe:"exename
Process name of the application to open

"tel:"telephone number
Places a telephone call.

"mailto:"email address
Starts the mailer.

"nexthead"
To the next chapter

"prevhead"
To the previous chapter

"next"
To the next page

"prev"
To the previous page

"top"
To the top

"bottom"
To the end

"back"
Back

"forward"
Forward (in history)

<map>
Defines an image map.
Possible child elements: block-level elements, area
The <map> tag has the following attributes.

[Attributes]

name [optional]
Map name
This attribute takes the following value(s).
Text

id [optional]
Gives an identifier to the element (image map). Must be unique in the XML file. Given in text data.

title [optional]
Map title
This attribute takes the following value(s).
Text

<link> (empty tag)
Defines document associations. Child element of <head>.
Possible child elements: none
The <link> tag has the following attributes.

[Attributes]

rel [optional]
Describes the relationship with the destination document as seen from this document. Multiple items may be specified.
This attribute takes the following value(s).
"stylesheet"
Denotes the style sheet applied by default.
"alternate stylesheet"
Denotes a style sheet not applied by default.

href [optional]
URI of the associated document (style sheet file name, etc.)

type [optional]
MIME type
Remarks: Only "text/css" may be used

media [optional]
The media envisioned as the output target. For screen, the screen size and number of colors can be specified.

<ruby>
Ruby characters
Possible child elements: The following combination of tags; (rb , rt), (rb, rp, rt, rp), (rbc, rtc) or (rbc, rp, rtc, rp) The order of these tags shall be observed.

<rb>
Text to display with ruby characters
Possible child elements: text data, inline elements, object elements

<rt>
Ruby character text
Possible child elements: text data, inline elements, object elements

<rbc>
Characters to apply ruby characters (encloses rb as a group)
Possible child elements: rb

<rtc>
Reading (encloses rt as a group)
Possible child elements: rt

<rp>
Possible child elements: text data, external_char
Specifies parentheses for alternate characters.
Coding example:

```
<ruby><rbc><rb>                                             被</rb><rb>ル</
rtc><rt>                                                    ひ</rt><rt>る</r
b>                                    漢字</rb><rp>（</rp><rt>かんじ</rt><rp>）</rp></ru
```

```
<ruby><rb>                                          漢字</rb>
<ruby><rb>                                              漢
</rp></ruby>
```

Remarks: The following position was adopted in the development of this standard.

- For single ruby characters, group ruby characters, separate coding methods are unnecessary.
- For ruby characters for idioms, it should be possible to insert separations into the ruby text.
- <rp> was left in as a part of ESP format (when converting to a distribution format, we envision this will normally be converted).
- Parentheses can also be used in the ruby characters without using <rp>.

<audio> (empty tag)
Plays audio data.
Possible child elements: none
Remarks: HTML5 compliant
The <audio> tag has the following attributes.

[Attributes]

src [optional]
Audio data file name

<video> (empty tag)
Plays video data.
Possible child elements: none
Remarks: HTML5 compliant
The <video> tag has the following attributes.

[Attributes]

src [optional]
Video data file name

<action> (empty tag)
Controls page turning between flows.
Possible child elements: none
The <action> tag has the following attributes.

[Attributes]

onopen [optional]
Specifies the file to execute when the page is opened.
This attribute takes the following value(s).
URI

onclose [optional]
Specifies the file to execute when the page is closed.
This attribute takes the following value(s).
URI

background [optional]
Execute in the background
This attribute takes the following value(s).
bool

cmd [optional]
Specifies the protocol.
Remarks: Independently added: Place a telephone call, start mailer
This attribute takes the following value(s).

"appl:"creator
Creator of the application to open

"exe:"exename
Process name of the application to open

"tel:"telephone number
Places a telephone call.

"mailto:"email address
Starts the mailer.

"nexthead"
To the next chapter

"prevhead"
To the previous chapter

"next"
To the next page

"prev"
To the previous page

"top"
To the top

"bottom"
To the end

"back"
Back

"forward"
Forward (in history)

<case/nocase>
Environment dependent operation (If a condition holds, the representation specified by <case> is played. When it doesn't, that specified by <nocase> is played.)
Possible child elements: text data, block-level elements, inline elements, object elements
Remarks: Enclose the section you wish to be the subject of selective operation with <case>...</case> or <nocase>...</nocase>.
The <case/nocase> tag has the following attributes.

[Attributes]
dir [optional]
Vertical text/horizontal text
This attribute takes the following value(s).
"vertical"
Vertical text
"horizontal"
Horizontal text
cpu [optional]
CPU
This attribute takes the following value(s).
CPU names such as "ppc", "68k", "x86." as conditions.
os [optional]
Operating system
This attribute takes the following value(s).
Names of Operating systems, such as "mac", "win32", "ios", "android" as conditions

screen [optional]
Window
This attribute takes the following value(s).
"pda"
The short side of the window size is 320 pixels or smaller.
"portrait"
The window proportion is vertically long (also true for squares).
"landscape"
The window proportion is horizontally long.

<mask>
Each time the text enclosed with the <mask> tag is clicked or the specified text is clicked, a mask can be applied or removed.
Possible child elements: text data, inline elements, object elements
The <mask> tag has the following attributes.

[Attributes]
initial_flag [optional]
Sets the initial state.
This attribute takes the following value(s).
"on"
Sets the masked state as the initial state.
"off"
Sets the unmasked state as the initial state.
trigger_id [optional]
Trigger for switching. Multiple triggers may be specified with "," as the separator.
id [optional]
Gives an identifier to the element. Must be unique in the XML file. Given in text data.
mask_type [optional]
Specifies the masking method. Either of the attribute values below may be specified.
This attribute takes the following value(s).
"default"
Masking with the viewer's default method.

"color"
Sets the text color and background color to the color specified by the color property. If the color property is omitted, the mask color is set to the value applied for text when the color property is omitted.

hold_flag [optional]
Specifies the method for saving the mask on/off state.
This attribute takes the following value(s).
"scope"
Saves the mask state until this text object entity is done being displayed.
"on_power"
Saves the mask state until the book is closed.
"save"
Saves the current mask state when the book is closed. Uses that state as the initial value when the book is next read.

class [optional]
Gives a grouping to the element. Given in text data.

<tts>
Segment information when TTS is reading the text
Possible child elements: text data, block-level elements, inline elements, object elements

Coding example: <tts>The computer is an important research tool</tts>
Remarques: The default language can be specified with <language>, which is a child element of <bibliography>, or with the xml:lang attribute of the <html> tag. Alternatively, place <span> elements in the <tts> element and the language can be expressed with the xml:lang attribute of <span>.
The <tts> tag has the following attributes.

[Attributes]
speaker [optional]
Speaker information
This attribute takes the following value(s).
"male"
Adult, male
"female"
Adult, female
"child"
Child

C.3.4 Dictionary-related tags

<ref>
Indicates a reference. It means "See also ...". Child element of <body>.
Possible child elements: text data, external_char
The <ref> tag has the following attributes.

[Attributes]
id [required]
Reference item ID. Specify an ID defined by the <dict_item> tag.
type [optional]
Reference type

<dict_item>
The tag that compiles the headword
Possible child elements: text data, inline elements, object elements, gender, psp, glabel, pronunciation, inflec, lang, slabel, spellout, variant, etymology
The <dict_item> tag has the following attributes.

[Attributes]
type [optional]
Type
id [required]
ID number
Remarks: This is usually required for editing purposes and doesn’t change when content is revised.

rank [optional]
Information such as an important word
level [optional]
Hierarchy level
page_break [optional]
Select whether or not to break the page at the end of this tag.
This attribute takes the following value(s).
bool
turning_page_control [optional]
If the page_break attribute is specified as "yes", this controls whether or not to prohibit navigating to the forward or back data from this word's page break position during the normal scrolling operation.
This attribute takes the following value(s).
"on"
The forward and back data cannot be navigated to.
"off"
Both the forward and back can be navigated to.
"forward"
The forward data can be navigated to, but the back data cannot be navigated to.
"back"
The back data can be navigated to, but the forward data cannot be navigated to.
revision [optional]
Indicates the revision of the content when adding this word data.
This attribute takes the following value(s).
Numeral
delete [optional]
Specifies whether or not to delete the content of this word data.
This attribute takes the following value(s).
bool

<reghead>
The tag that compiles the search term. This tag has one or more <headword> and zero or more <key> child elements.
Possible child elements: headword, key

<headword>
This tag exists for each search term.
Possible child elements: text data, external char
Remarque: One or more exist in child elements for head.
The <headword> tag has the following attributes.
[Attributes]
type [optional]
Search term type. For pronunciation, "pronunciation".
This attribute takes the following value(s).
"pronunciation"
Pronunciation
table_id [required]
Specifies the search table.
phonetic_notation [optional]
Specify the phonetic notation to display the language sounds as text.
The default is "IPA".
This attribute takes the following value(s).
"IPA"
The phonetic notation is the International Phonetic Alphabet.

<key>
Input text for search. In addition to hiragana and alphabetic characters, code may be entered that contains standard characters.
Possible child elements: text data
The <key> tag has the following attributes.

[Attributes]
type [optional]
Sets the type of characters permitted in the search table.

<meaning>
Word meaning. (Set for every meaning number.)
Possible child elements: inline elements, object elements
The <meaning> tag has the following attributes.
[Attributes]
type [optional]
Category
subid [optional]
The reference ID that references the position in the definition
level [optional]
Meaning level. Specify as a number 0 to 9.
no [optional]
The number of meanings for that level. Specify as a number 0 to 9.

<example>
Example
Possible child elements: inline elements, object elements
The <example> tag has the following attributes.
[Attributes]
type [optional]
Example type

<subhead>
Subheading start (not required)
Possible child elements: subheadword, meaning, example, key
The <subhead> tag has the following attributes.
[Attributes]
subid [optional]
Relevant subheading reference ID
type [optional]
Subheading type

<subheadword>
Subheadword
Possible child elements: text data, external char
The <subheadword> tag has the following attributes.
[Attributes]
subid [optional]
Relevant subheadword reference ID
type [optional]
Subheadword type

<split>
Splits data into alphabetic or other units, an typical example of which is grouping entries by the first letter.
Possible child elements: inline elements, object elements
The <split> tag has the following attributes.
[Attributes]
level [optional]
Hierarchy level. Specify as a number 0 to 9.

<column>
Creates a boxed column or shaded column.
Possible child elements: inline elements, object elements
The <column> tag has the following attributes.
[Attributes]
subid [required]
Reference ID of the relevant column

type [optional]
Column type

<gender>
Gender type. Child element of <dict_item>.
Possible child elements: inline elements, object elements

<glabel>
Grammar label. Child element of <dict_item>.
Possible child elements: inline elements, object elements

<pronunciation>
Phonetic symbols. Child element of <dict_item>.
Possible child elements: inline elements, object elements
The <pronunciation> tag has the following attributes.
[Attributes]
phonetic_notation [optional]
Specify the phonetic notation to display the language sounds as text.
The default is "IPA".
This attribute takes the following value(s).
"IPA"
The phonetic notation is the International Phonetic Alphabet.

<psp>
Part of speech. Child element of <dict_item>.
Possible child elements: inline elements, object elements

<inflec>
Inflection. Child element of <dict_item>.
Possible child elements: inline elements, object elements

<lang>
Language type. Child element of <dict_item>.
Possible child elements: inline elements, object elements

<slabel>
Technical terminology label. Child element of <dict_item>.
Possible child elements: inline elements, object elements

<spellout>
Abbreviated portion of the headword. Child element of <dict_item>.
Possible child elements: inline elements, object elements
The <spellout> tag has the following attributes.
[Attributes]
org [optional]
What manner to replace in the display

<variant>
Variant word. Child element of <dict_item>.
Possible child elements: inline elements, object elements

<search_page>
Search page object entity
Possible child elements: search_page_title, key_input_region, search_link_item

<search_page_title>
Search page title and image
Possible child elements: text data, external_char
The <search_page_title> tag has the following attributes.
[Attributes]
src [optional]
Search page image data file name

<key_input_region>

Headword input region

Possible child elements: key_input_region_prompt, enable_key_type

The <key_input_region> tag has the following attributes.

[Attributes]

table_id [required]

The search table ID to add for handling the headword input region

This attribute takes the following value(s).

Text

search_type [optional]

Specifies the type of search.

This attribute takes the following value(s).

"matches_only"

Refining search. The text entered in the entry word input region is compared from the starting character and only headwords that match that portion are displayed as search results.

"matches_first"

First match search. The text entered in the entry word input region is compared from the starting character and the headword that matches that portion and the headwords registered after that headword are displayed as search results.

<key_input_region_prompt>

Text displayed near the input region. Child element of <key_input_region>

Possible child elements: text data, external_char.

<enable_key_type>

Defines what types of characters are allowed in the keyword input field. Cannot be omitted.The attributes and child elements of this tag are the same as those used when this tag is a child of <search_table_def>. The character types allowed in the tag when it is a child of <key_input_region> shall be a subset of those defined in the associated search table. Note that this means it is allowed for the former to prohibit character types enabled by the latter, whereas the opposite is not allowed.

Possible child elements: char_list

<search_link_item>

Sets information related to the link list.

Possible child elements: search_link_title

The <search_link_item> tag has the following attributes.

[Attributes]

id [required]

Text ID

This attribute takes the following value(s).

Text

<search_link_title>

Title to display on the link list. Child element of <search_link_item>.

Possible child elements: text data, external_char.

The <search_link_title> tag has the following attributes.

[Attributes]

xml:lang [optional]

Default language

lang_code [optional]

Specifies the language code for the title.

<etymology>

Etymology information. Child element of <dict_item>

Possible child elements: inline elements, object elements.

<tts>

Segment information when TTS is reading the text

Possible child elements: text data, block-level elements, inline elements
Remarks: Coding example: <tts>The computer is an important research tool</tts> The default language can be specified with <language>, which is a child element of <bibliography>, or with the xml:lang attribute of the <html> tag. In addition, if the default language is a child element of <dict_item>, the default language can be specified with the lang attribute of <dict_item>. In other cases, place <span> elements in the <tts> element and the text is expressed with that lang attribute.
The <tts> tag has the following attributes.

[Attributes]

speaker [optional]
Speaker information
This attribute takes the following value(s).
"male"
Adult male
"female"
Adult female
"child"
Child

C.3.5 Search-table related tags

<search_table>

Root element for the search table description. At most only one instance of the tag happens in content.
Possible child elements: search_table_def
The <search_table> tag has the following attributes.

[Attributes]

bookmark [optional]
Specifies if all the search tables included in the content should be eligible for automatic bookmarking on the viewer (i.e. automatically keeping the tally of the headwords searched). The default is “false”.
This attribute takes the following value(s).
bool

wordbook [optional]
Specifies if all the search tables are eligible for the wordbook function on the viewer (i.e. automatically registering headwords in the wordbook and showing the list). The following values are possible. The default is “false”.
This attribute takes the following value(s).
bool

jump_search_root [optional]
Specifies if the content is eligible as a source of multi-content search from text (i.e. can launch multi content search using the keyword selected from the displayed text of the content). The following values are possible. The default is “true”.
This attribute takes the following value(s).
bool

jump_search [optional]
Specifies if all the search tables are eligible for multi-content search from text (i.e. can be searched as one of the contents in multi-content search using the keyword selected from the text) and multi-content search from input with the content being displayed (i.e. can be searched as one of the contents in multi-content search using the input keyword) with the content displayed. The default is “true”.
This attribute takes the following value(s).
bool

all_search [optional]
Specifies if all the search tables are eligible for multi-content search without the content itself being displayed. The default is “true”.
This attribute takes the following value(s).
bool

<search_table_def>

Specifies information concerning each search table. There shall be at least one instance of this tag as a child element of <search_table>.
Possible child elements: enable_key_type, key_normalization
The <search_table_def> tag has the following attributes.

[Attributes]

id [required]
Defines the ID number of this search table. The ID number shall be unique to each search table within the content.
This attribute takes the following value(s).
Text data

use_default [optional]
Defines whether a search based on this search table is allowed by the viewer, when the search is not initiated from a related search page. The default is “false”.
This attribute takes the following value(s).
bool

sorting_rule [optional]
Defines how to sort the search results. This attribute is to allow for different ordering schemes for different languages. The default is "implicit".
This attribute takes the following value(s).
"implicit"
Ordered as written in the content.
"unicode"
Ordered by Unicode value.
"other"
In the other manner not expressed.

name [optional]
Sets the name for the search table. in standard character string. The name is intended to be used in listing the search results.

short_name [optional]
Sets the shortened version of the name of the table. Names longer than the one specified by the name attribute is not allowed. The name is intended to be used in listing the search results.

wild [optional]
Specifies whether table for wildcard search should be generated. The default is “false”.
This attribute takes the following value(s).
bool

blank [optional]
Specifies whether search table for “blank word” search should be generated. The default is “false”.
This attribute takes the following value(s).
bool

end [optional]
Specifies whether search table for “word-ending” search should be generated. The default is “false”.
This attribute takes the following value(s).
bool

help_item_id [optional]
Sets the shortened version of the name of the table. Names longer than the one specified by the name attribute is not allowed. The name is intended to be used in listing the search results.

<enable_key_type>

Defines the type of characters that can be used to store the lookup key of the entry words in the table. A child element of <search_table_def>.
Possible child elements: char_list
The <enable_key_type> tag has the following attributes.

[Attributes]

numerals [optional]
Defines whether the lookup key may include numerals or not. The default is “false”.
This attribute takes the following value(s).

bool

basic_alphabet [optional]

Defines whether the lookup key may include alphabet (limited to the non-accentuated characters found in US-ACSII) or not. The default is “false”.

This attribute takes the following value(s).

bool

kana [optional]

Defines whether the lookup key may include Japanese hiragana and katakana. The default is “false”.

This attribute takes the following value(s).

bool

kanji [optional]

Defines whether the lookup key may include kanji. The default is “false”.

This attribute takes the following value(s).

bool

listed [optional]

The characters available for the lookup keys are given by list. The list is given by < char_list> tag when true and the numerals/basic_alphabet/kana/kanji attributes are ignored. The default is “false”.

This attribute takes the following value(s).

bool

<key_normalization> (empty tag)

Defines the normalization methods to be applied on the keys registered in this search table. A child element of <search_table_def>.

Possible child elements: none

The <key_normalization> tag has the following attributes.

[Attributes]

capitalization [optional]

Specifies if all (alphabetical) characters in the table are converted to upper case. The default is “true”.

This attribute takes the following value(s).

bool

cho_on [optional]

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".

This attribute takes the following value(s).

"delete"

Delete cho_on

"repeat"

Repeat the vowel preceding the character

"as_is"

No conversion is done

daku_on [optional]

Specifies if Japanese voiced sound (daku on) conversion is done. The default is “true”.

This attribute takes the following value(s).

bool

handaku_on [optional]

Specifies if Japanese voiced sound (han_daku on) conversion is done. The default is “true”.

This attribute takes the following value(s).

bool

soku_on [optional]

Specifies if Japanese geminate consonant (soku on or "small tsu") conversion is done. The default is “true”.

This attribute takes the following value(s).

bool

yo_on [optional]

Specifies if Japanese palatalization (yo on) conversion is done. The default is “true”.

This attribute takes the following value(s).
bool

other_small_kana [optional]

Specifies if Japanese small character (hiragana and katakana) conversion is done. The default is “true”.
This attribute takes the following value(s).
bool

<char_list>

Specifies a character list when listed="true" in <enable_key_type>. A child element of <enable_key_type>.
Possible child elements: text data

C.3.6 Comic-related tags

<comic_cell_type_body>

The parent tag for comic body data for displaying cells. Child element of <body>. This tag has one or more <cell> child elements.
Possible child elements: cell
The <comic_cell_type_body> tag has the following attributes.

[Attributes]

bg_color [optional]

The background color in the image display region when displaying cell images in this comic body data for displaying cells

bg_color_space [optional]

Color space for the background color
This attribute takes the following value(s).
"RGB"
The color space is RGB.

<comic_page_type_body>

The parent tag for comic body data used displaying pages. Child element of <body>. This tag has one or more <page> child elements.
Possible child elements: page
The <comic_page_type_body> tag has the following attributes.

[Attributes]

bg_color [optional]

The background color in the image display region when displaying the page images in this comic body data for displaying pages

bg_color_space [optional]

Color space for the background color
This attribute takes the following value(s).
"RGB"
The color space is RGB.

order [optional]

For each page of comic body data for displaying pages, specifies the display order (start position and end position) of frames (equivalent to cell images) on the page.
This attribute takes the following value(s).

"RT-LB"

Top right start – bottom left end. When switching pages and turning to the next page, the page image is displayed by matching the top right edge of the page image to the top right edge of the image display region. When turning to the previous page, the page image is displayed by matching the bottom left edge of the page image to the bottom left edge of the image display region.

"LT-RB"

Top left start – bottom right end. When switching pages and turning to the next page, the page image is displayed by matching the top left edge of the page image to the top left edge of the image display region. When turning to the previous page, the page image is displayed by matching the bottom right edge of the page image to the bottom right edge of the image display region.

"RB-LT"

Bottom right start – top left end. When switching pages and turning to the next page, the page image is displayed by matching the bottom right edge of the page image to the bottom right edge of the image display region. When turning to the previous page, the page image is displayed by matching the top left edge of the page image to the top left edge of the image display region.

"LB-RT"

Bottom left start – top right end. When switching pages and turning to the next page, the page image is displayed by matching the bottom left edge of the page image to the bottom left edge of the image display region. When turning to the previous page, the page image is displayed by matching the top right edge of the page image to the top right edge of the image display region.

<cell>

Cell data. This tag can be a child element of the a tag and it can specify a map as a child element.

Possible child elements: cell_scene, speech, map

The <cell> tag has the following attributes.

[Attributes]

src [required]

This cell's image

bg_color [optional]

The background color in the image display region

bg_color_space [optional]

Color space for the background color

This attribute takes the following value(s).

"RGB"

The color space is RGB.

fit [optional]

Sets whether or not to permit the fit display (preserves the aspect ratio) of the scene when the terminal's display screen size is smaller than the recommended screen size (equivalent to the scene size) specified by the display_size attribute in the <bvf> tag. The default is "false".

This attribute takes the following value(s).

bool

effect [optional]

Specifies the display effect when displaying this cell image from the previous cell image.

This attribute takes the following value(s).

"fade-in"

After displaying the image display region in the background color, the cell image is gradually displayed.

"dissolve"

The next cell image is gradually displayed on top of the previous cell.

"slide-right"

After displaying the image display region in the background color, the cell image is displayed sliding from left to right.

"slide-left"

After displaying the image display region in the background color, the cell image is displayed sliding from right to left.

"slide-up"

After displaying the image display region in the background color, the cell image is displayed sliding from the bottom upward.

"slide-down"

After displaying the image display region in the background color, the cell image is displayed sliding from the top downward.

"overwrite-right"

The next cell image is displayed on top of the previous cell by overwriting it from left to right.

"overwrite-left"

The next cell image is displayed on top of the previous cell by overwriting it from right to left.

"overwrite-up"
The next cell image is displayed on top of the previous cell by overwriting it from the bottom upward.

"overwrite-down"
The next cell image is displayed on top of the previous cell by overwriting it from the top downward.

"box-center"
After displaying the image display region in the background color, the cell image is displayed by expanding as a square in four directions from the center.

"box-rightdown"
After displaying the image display region in the background color, the cell image is displayed by expanding as a square from the top left to the bottom right.

"box-leftdown"
After displaying the image display region in the background color, the cell image is displayed by expanding as a square from the top right to the bottom left.

"box-rightup"
After displaying the image display region in the background color, the cell image is displayed by expanding as a square from the bottom left to the top right.

"box-leftup"
After displaying the image display region in the background color, the cell image is displayed by expanding as a square from the bottom right to the top left.

"box-free"
After displaying the image display region in the background color, the cell image is displayed by expanding as a square from the desired cell image position. The start position for the expanding box display is specified with the effect_pos attribute.

"pushout-right"
The next cell image is displayed while pushing out the previous cell image from left to right.

"pushout-left"
The next cell image is displayed while pushing out the previous cell image from right to left.

"pushout-up"
The next cell image is displayed while pushing out the previous cell image from the bottom upward.

"pushout-down"
The next cell image is displayed while pushing out the previous cell image from the top downward.

"wipe-right"
The previous cell image is displayed while changing to the next cell image from left to right.

"wipe-left"
The previous cell image is displayed while changing to the next cell image from right to left.

"wipe-up"
The previous cell image is displayed while changing to the next cell image from the bottom upward.

"wipe-down"
The previous cell image is displayed while changing to the next cell image from the top downward.

"wipe-vertical-in"
The previous cell image is displayed while changing to the next cell image from the top and bottom inward.

"wipe-vertical-out"

The previous cell image is displayed while changing to the next cell image from the center out to the top and bottom.

"wipe-horizontal-in"

The previous cell image is displayed while changing to the next cell image from the right and left inward.

"wipe-horizontal-out"

The previous cell image is displayed while changing to the next cell image from the center out to the right and left.

"wipe-center-in"

The previous cell image is displayed while changing to the next cell image from the four corners inward.

"wipe-center-out"

The previous cell image is displayed while changing to the next cell image from the center out to the four corners.

"randomblock"

The next cell image is gradually displayed on top of the previous cell randomly in blocks.

effect_time [optional]

Sets the time for the display effect specified by the effect attribute as the relative time in 10 steps from 1 (fast) to 10 (slow).

effect_pos [optional]

Specifies the start position when specifying the box enlarging display from the desired position as the display effect ("box-free" is specified in the effect attribute). This attribute shall not be omitted when "box-free" is specified in the effect attribute. This attribute is ignored when any other value is specified. The range for the start position that can be specified is set to within the scene.

scroll [optional]

Specifies the cell scrolling method.

This attribute takes the following value(s).

"custom"

Sets the scene to the desired coordinate position for the cell image (specified by the position attribute of the <cell_scene/> tag) and scrolls between the set scene.

"up"

Scrolls the cell image from the bottom edge to the top edge.

"down"

Scrolls the cell image from the top edge to the bottom edge.

"left"

Scrolls the cell image from the right edge to the left edge.

"right"

Scrolls the cell image from the left edge to the right edge.

"no"

No scrolling (the center coordinates of the cell image are set and displayed in the center of the scene.).

scroll_time [optional]

Sets the scroll time for this cell as the relative time in 10 steps from 1 (fast) to 10 (slow).

vibration [optional]

Sets whether or not to vibrate the terminal when this cell is displayed.

This attribute takes the following value(s).

bool

display_vibration [optional]

Sets whether or not to vibrate the screen when this cell is displayed.

This attribute takes the following value(s).

bool

vibration_direction [optional]

Sets the screen vibration direction.

This attribute takes the following value(s).

"vertical"

Vertical direction

"horizontal"

Horizontal direction

vibration_time [optional]
Sets the terminal vibration time and the screen vibration time as the relative time in 5 steps from 1 (short) to 5 (fast).

backlight [optional]
Sets whether or not to flash the backlight when this cell is displayed.
This attribute takes the following value(s).
bool

backlight_time [optional]
Sets the number of times to flash the backlight in 5 steps from 1 to 5.

backlight_cycle [optional]
Sets the backlight flashing cycle as the relative time in 3 steps from 1 (short) to 3 (fast).

sound [optional]
Specifies the sound effect when displaying this cell.

url_jump [optional]
Specifies the URL address for the web site (restricted to html) associated with this cell.

jump_itemid [optional]
Specifies the jump destination page ID when switching from the cell display to the page display.

usemap [optional]
Specifies a client-side image-map.

id [optional]
Gives an identifier to the element. Must be unique in the XML file. Given in text data. Must be given uniquely to each <cell> and <page> tag in the XML file.

<cell_scene>

Cell scene data. Child element of <cell>.
Possible child elements: cell_draw_image
The <cell_scene> tag has the following attributes.

[Attributes]

position [optional]
Specifies the coordinates for the cell image to display in the center of the image display area in this scene.

scroll_time [optional]
Sets the scroll time when navigating from this scene to the next scene as the relative time in 10 steps from 1 (fast) to 10 (slow).

draw_image_time [optional]
Sets the time after displaying the scene and until displaying the overlay image as the relative time in 3 steps from 1 (short) to 3 (fast).

vibration [optional]
Sets whether or not to vibrate the terminal when this scene is displayed.
This attribute takes the following value(s).
bool

display_vibration [optional]
Sets whether or not to vibrate the screen when this scene is displayed.
This attribute takes the following value(s).
bool

vibration_direction [optional]
Sets the screen vibration direction.
This attribute takes the following value(s).
"vertical"
Vertical direction
"horizontal"
Horizontal direction

vibration_time [optional]
Sets the terminal vibration time and the screen vibration time as the relative time in 5 steps from 1 (short) to 5 (fast).

backlight [optional]
Sets whether or not to flash the backlight when this scene is displayed.
This attribute takes the following value(s).
bool

backlight_time [optional]
Sets the number of times to flash the backlight in 5 steps from 1 to 5.
backlight_cycle [optional]
Sets the backlight flashing cycle as the relative time in 3 steps from 1 (short) to 3 (fast).
sound [optional]
Specifies the sound effect when displaying this scene.
url_jump [optional]
Specifies the URL address for the web site (restricted to html) associated with this scene.

<cell_draw_image> (empty tag)
The image data to paste in the cell. Child element of <cell_scene>.
Possible child elements: none
The <cell_draw_image> tag has the following attributes.
[Attributes]
src [required]
Specifies the overlay image to display on the cell image.
position [required]
The display position (coordinates) on the cell image for the overlay image specified by the src attribute.
speech [optional]
Speech (text).
speaker [optional]
Speaker. The name of the character that speaks the speech.

<page>
Page data. Child element of <comic_page_type_body>.
Possible child elements: page_image, speech
The <page> tag has the following attributes.
[Attributes]
bg_color [optional]
The background color in the image display region when displaying the image for this page (specified by the <page_image/> tag).
bg_color_space [optional]
Color space for the background color
Example: "RGB"
The color space is RGB.
id [required]
Gives an identifier to the element. Must be unique in the XML file. Given in text data. Must be given uniquely to each <cell> and <page> tag in the XML file.
cell_jump_id [required]
Specifies the jump destination cell ID when switching from the page display to the cell display.

<page_image> (empty tag)
Page image data. Child element of <page>.
Possible child elements: none
The <page_image> tag has the following attributes.
[Attributes]
src [required]
Specifies this page's image.

<speech>
Speech. Child element of <cell> and <page>. Write the text in the child element. May be specified multiple times. May be omitted.
Possible child elements: text data
The <speech> tag has the following attributes.
[Attributes]
speaker [optional]
Speaker. The name of the character that speaks the speech.

C.3.7 Bibliography data tags

<bibliography>

The root element of the bibliography information.

Possible child elements: title, edition, creator, subject, description, publisher, distributor, contributor, date, identifier, source, language, relation, coverage, rights, price, comment, classification, rating, synopsis, image

<title>

Defines the title. Multiple tags may be written. Child element of <bibliography>.

Possible child elements: external_char, ruby, comment

The <title> tag has the following attributes.

[Attributes]

reading [optional]

The reading of the title.

type [optional]

Title type

This attribute takes the following value(s).

"title"

Title name

"series"

Series name

"subtitle"

Subtitle name

"other"

Other

<edition>

Edition. Child element of <bibliography>.

Possible child elements: external_char, ruby, comment

The <edition> tag has the following attributes.

[Attributes]

reading [optional]

Reading

<creator>

Creator. Defines details by having <organization>, <person>, and <comment> for child elements. Child element of <bibliography>.

Possible child elements: organization, person, comment

The <creator> tag has the following attributes.

[Attributes]

role [optional]

Role

This attribute takes the following value(s).

"designer"

Designer

"author"

Author

"editor"

Editor

"translator"

Translator

"supervisor"

Supervisor

"photographer"

Photos

"illustrator"

Illustrations

"binder"

Binder

"planner"

Planner

"other"
Other

<subject>
Theme. Child element of <bibliography>.
Possible child elements: keyword, comment

<description>
Details. Child element of <bibliography>.
Possible child elements: text data, external_char, ruby, comment

<publisher>
Registers publishing company information. Defines details by having <organization>, <person>, <comment>, and <code> for child elements. Child element of <bibliography>.
Possible child elements: organization, person, comment, code
The <publisher> tag has the following attributes.
[Attributes]
country [optional]
Country of publication. Enter the A3 country code defined by ISO 3166-1 in lowercase. For example, Japan is "jpn".

<code>
The code for the publishing company or other entity. Child element of <publisher> and <distributor>.
Possible child elements: text data, external_char, ruby, comment
The <code> tag has the following attributes.
[Attributes]
type [optional]
Publishing company code type. The content may be freely written.

<distributor>
Registers distributor information. Defines details by having <organization>, <person>, <comment>, and <code> for child elements. Child element of <bibliography>.
Possible child elements: organization, person, comment, code

<contributor>
Registers contributor information. Defines details by having <organization>, <person>, and <comment> for child elements. Child element of <bibliography>.
Possible child elements: organization, person, comment
The <contributor> tag has the following attributes.
[Attributes]
role [optional]
Role

<date>
Date. Holds the date written according to the standard specified by the system attribute. Child element of <bibliography>.
Possible child elements: text data, external-char, ruby, comment
The <date> tag has the following attributes.
[Attributes]
type [optional]
Publication date/sales date/revision date type
This attribute takes the following value(s).
"publication"
Publication date
"sale"
Sales date
"revision"
Revision date
system [optional]
Specifies the standard for writing the date. The default is "ISO 8601".

<identifier>

Defines the content identifier. Child element of <bibliography>.
Possible child elements: text data, external-char, ruby, comment
The <identifier> tag has the following attributes.

[Attributes]
type [required]
Specifies the type of identifier. Example: "ISBN"

<source>

Reference to the deriving source. Child element of <bibliography>.
Possible child elements: text data, external-char, ruby, comment

<language>

Language. Child element of <bibliography>.
Possible child elements: text data
The <language> tag has the following attributes.

[Attributes]
system [optional]
The standard that defines the language names.

<relation>

Reference to related information. Child element of <bibliography>
Possible child elements: text data, external-char, ruby, comment
.

<coverage>

Range or target. Child element of <bibliography>.
Possible child elements: text data, external-char, ruby, comment

<rights>

Information related to rights. Child element of <bibliography>.
Possible child elements: text data, external-char, ruby, comment

<price>

Price. Child element of <bibliography>.
Possible child elements: text data, comment
The <price> tag has the following attributes.

[Attributes]
currency [required]
Currency
Remarks: Conform to ISO 4217 currency codes. For example, the Japanese yen is JPY.
country [optional]
Target country
Remarks: Enter the A3 country code defined by ISO 3166-1 in lowercase. For example, Japan is "jpn".

<person>

Defines information on individuals as a child element of <publisher>, <distributor>, and <contributor>. Defines names, contact information, and comments by having <name>, <contact>, and <comment> tags as child elements.
Possible child elements: name, contact, comment

<organization>

Organization information. Defines organization information as a child element of <publisher>, <distributor>, and <contributor>. Defines names, contact information, and comments by having <name>, <contact>, and <comment> tags as child elements.
Possible child elements: name, contact, comment

<name>

Name. Child element of <person>, <organization>, and <address>. May be specified multiple times.

Possible child elements: text data, external-char, ruby, comment
The <name> tag has the following attributes.

[Attributes]

type [optional]

Name type. May be omitted when the direct parent tag of the <name> tag is <organization>. This tag also expresses place names.
This attribute takes the following value(s).
"first"
First name
"middle"
Middle name
"last"
Last name
"place"
Location name (place name)
"other"
Other name

reading [optional]

Reading

<contact>

Information other than the name. Child element of <person> and <organization>.
Possible child elements: address, email, telephone, fax, website, comment

<address>

Address. Child element of <contact>. May be specified multiple times.
Possible child elements: name, postcode, comment
The <address> tag has the following attributes.

[Attributes]

reading [optional]

Address reading.

<comment>

Comment
Possible child elements: text data

<classification>

Classification
Possible child elements: text data,comment
The <classification> tag has the following attributes.

[Attributes]

type [required]

Classification type
Remarks: Japanese_C_CODE

<rating>

Rating. Sets the adult designation and violence designation.
Possible child elements: text data, comment
The <rating> tag has the following attributes.

[Attributes]

type [optional]

Rating type
This attribute takes the following value(s).
"adult"
Subject to an adult designation
"violent"
Subject to a violence designation

<synopsis>

Summary
Possible child elements: text data, external_char, ruby, comment

<keyword>

Registers a keyword. Child element of <subject>.
Possible child elements: text data, external_char, ruby, comment
The <keyword> tag has the following attributes.

[Attributes]
type [optional]
Keyword category
reading [optional]
Reading

<image>

Specifies an image file name in the child element and defines that type. Child element of <bibliography>.
Possible child elements: text data
The <image> tag has the following attributes.

[Attributes]
type [optional]
Image type
This attribute takes the following value(s).
"spine"
Spine
"front"
Front cover
"thumbnail"
Thumbnail
"other"
Other

<postcode>

Postal code. Child element of <address>. Only one tag may be specified. May be omitted.
Possible child elements: text data, comment

<email>

Email address. Child element of <contact>. May be specified multiple times.
Possible child elements: text data, comment

<telephone>

Telephone number. Child element of <contact>. May be specified multiple times.
Possible child elements: text data, comment

<fax>

Fax number. Child element of <contact>. May be specified multiple times.
Possible child elements: text data, comment

<website>

Website address. Child element of <contact>. May be specified multiple times.
Possible child elements: text data, comment

C.3.8 Global settings data tags

<global_setting>

Root tag
Possible child elements: default_ccs, page_progression_direction, window, proprietary, permission

<default_ccs>

The character group name that indicates the range of standard characters and extended characters used in the content. Write in a child element. Do not enclose them with double quotes. If there are multiple items, connect them with a half-width comma ",". (Example: JIS X 0208:1997, JIS X 0213).

Possible child elements: text data

<page_progression_direction>

Defines the binding direction (the page progression direction). For right to left, write "rl" or "rtl" in a child element. For left to right, write "lr" or "ltr" in a child element.
Possible child elements: text data

<window> (empty tag)

Sets the window size.
Possible child elements: none
The <window> tag has the following attributes.
[Attributes]
width [optional]
Sets the window size (width).
This attribute takes the following value(s).
Numeric value (number of pixels)

height [optional]
Sets the window size (height).
This attribute takes the following value(s).
Numeric value (number of pixels)

<proprietary>

Enclose vendor-specific representations that need to be stored in the content data but that do not need to be displayed/converted.
Possible child elements: any element, as no checking is done for its child elements and down..
The <proprietary> tag has the following attributes.
[Attributes]
vendor [required]
A vendor-specific string such as the name of the vendor or the rendering system.
Example: ("ttime", "xmdf")

C.3.9 Table of contents/item data tags

<package>

Root element.
Remarks: <manifest>, <spine>, and <special_page_link> (as necessary) shall be defined as child elements in that order.
Possible child elements: manifest, spine, special_page_link
The <package> tag has the following attributes.
[Attributes]
version [optional]
Version

<manifest>

Describes the item list. Child element of <package>.
Remarks: Using <item>, define the item data used in the content in child elements in order.
Possible child elements: item
The <manifest> tag has the following attributes.
[Attributes]
version [optional]
Version

<item>

Indicates item data used in the content. Child element of <manifest>. This tag only has <permission> as a child element.
Possible child elements: permission
The <item> tag has the following attributes.
[Attributes]

id [required]
Text for identification
This attribute takes the following value(s).
Text
href [required]
File name
media-type [required]
References the "item" (manifest) file id.
Example
"application/xml"
Body data
"application/x-bvf-flip-animation"
Animation
Remarks: Apart from these two cases, it should be set according to the file.

<spine>
Indicates the order of the items. Child element of <package>.
Possible child elements: itemref
The <spine> tag has the following attributes.
[Attributes]
toc [optional]
Indicates which file is the table of contents. Specify an id defined by <item> in <manifest>.
global_setting [optional]
Indicates which file is the global settings file. Specify an id defined by <item> in <manifest>.
search_table [optional]
Indicates which file is the search table file. Specify an id defined by <item> in <manifest>.
bibliography [optional]
Indicates which file is the bibliography data file. Specify an id defined by <item> in <manifest>.

<itemref> (empty tag)
A child element of <spine>. This tag defines the order of each item contained in the content by arranging the tag in order.
Possible child elements: none
The <itemref> tag has the following attributes.
[Attributes]
idref [required]
Specify an id defined by <item> in <manifest>.

<special_page_link>
Using <special_page> in a child element, information such as flow position is registered in advance to make it possible to easily jump to frequently referenced pages and pages important for reading the book (for example, a character correlation chart page in novels or pages with maps in a travel guide). Child element of <package>.
Possible child elements: special_page

<special_page>
Special page information is coded for each position information to register by specifying an id registered in <item> in <manifest> or an object ID in the file for that <item>. Child element of <special_page_link>.
Possible child elements: text data
The <special_page> tag has the following attributes.
[Attributes]
type [optional]
Content type for the position information to register
The default is "other".
This attribute takes the following value(s).
"cover"
Front cover

"title_page"
Title page
"preface"
Preface
"contents"
Table of contents
"body"
Start of the body
"column"
Column
"note"
Note
"figure"
Figure
"ad"
Advertisement
"afterword"
Afterword
"appendix"
Appendix
"answer"
Answers
"glossary"
Glossary
"bibliography"
Bibliography
"commentary"
Commentary
"index"
Index
"imprint"
Imprint
"author_info"
Author introduction
"other"
Other
"flow_title"
Flow title

title [optional]
The title of the registered position information

C.3.10 Animation data tags

<flip_animation>

Still image sequence animation. To be written in an separate .xml file for each animation. Refer to the animation by specifying the name of the animation XML file in the src attribute of the <img> tag. To add audio to this animation, specify <audio> in a child element. This tag shall have one or more <flip_animation_source> tags in child elements.
Possible child elements: audio, flip_animation_source
The <flip_animation> tag has the following attributes.

[Attributes]

renewal_time [optional]
The interval to switch each still image (seconds or milliseconds). Write as "...s" or "...ms" in a child element.

<flip_animation_source> (empty tag)

Specifies a still image that makes up the animation. Child element of <flip_animation>.
Possible child elements: none
The <flip_animation_source> tag has the following attributes.

[Attributes]

src [required]
Still image file name

renewal_time [optional]

For the switching interval for each still image, in cases where you wish to specify a value that is different from the switching interval specified by the renewal_time attribute in the parent element <flip_animation> (seconds or milliseconds). Write as "...s" or "...ms" in a child element.

C.4 Style data

C.4.1 General

Style data consists of properties and their values applied to tags. The properties usable in style data are listed in C.4.2 while tag(selector)-property combinations that are allowed are listed in C.4.3.

C.4.2 Properties

Property: **font-family**

Specifies the font family.
This property may take the value(s) below as the property value.
Font name

Property: **font-style**

Specifies the font style.
This property may take the value(s) below as the property value.

"normal"
Specifies the standard font in the font family.

"italic"
Specifies the italic font in the font family (font of the type designed with an exclusive italic typeface).

"oblique"
Specifies oblique in the font family (font of the type designed to give the impression that the characters are simply slanted).

Property: **font-variant**

Displays the font as a small-caps font.
This property may take the value(s) below as the property value.

"normal"
Specifies the standard, not small-caps font.

"small-caps"
Specifies the small-caps font.

Property: **font-size**

Body text size
This property may take the value(s) below as the property value.

[n]px
Specifies the body text size as a numeric value. (unit: px)

[n]%, [n]em

"maximum"
Displays text with the maximum text size.

"big"
Displays text with the large text size.

"medium"
Displays text with the medium text size.

"small"
Displays text with the small text size.

"minimum"
Displays text with the minimum text size.

Property: **font-base**

The base when specifying the text size
This property may take the value(s) below as the property value.

"last"

Sets the last text as the base.
"default"
Sets the default text as the base.

Property: **color**
Specifies the color.
This property may take the value(s) below as the property value.
colorvalue

Property: **color-space**
Specifies the color space.
This property may take the value(s) below as the property value.
"RGB"
The color space is RGB.
"SRGB"
The color space is SRGB.

Property: **filter**
Adds a display effect to text or images.
This property may take the value(s) below as the property value.
"invert()"
Inverts the colors.

Property: **font-weight**
Font thickness
This property may take the value(s) below as the property value.
"normal"
Same as 400
"bold"
Same as 700
"bolder"
Thicker than the inherited value
"lighter"
Thinner than the inherited value
100
200
300
400
normal
500
600
700
bold
800
900
"inherit"

Property: **text-decoration**
Specifies underlining (when horizontal text) or sidelining (when vertical text).
This property may take the value(s) below as the property value.
"underline".
Displays underlining (when horizontal text) or sidelining (when vertical text).
"none"
Nothing

Property: **text-emphasis-style**
Emphasis (dots, small dots for emphasis)
Remarks: Attribute values have been combined with CSS Text Level 3 (Working Draft). These have been defined by JIS X 4052 except for none, open dot, and filled dot. For compatibility between formats and from trends in past printed materials, filled circle (primarily horizontal text) and filled sesame (primarily vertical text) should be used.

This property may take the value(s) below as the property value.

"none"
None

"filled sesame"
Sesame dot

"filled double-circle"
Fish eye

"filled dot"
Black dot

"filled circle"
Black circle

"filled triangle"
Black up-pointing triangle

"open sesame"
White sesame dot

"open double-circle"
Bull's eye

"open dot"
White dot

"open circle"
White circle

"open triangle"
White up-pointing triangle

Text
Use the specified text (enclose with double quotes) as emphasis dots.

Property: **text-shadow**

Applies a shadow to the text style.
This property may take the value(s) below as the property value.
[horizontal distance]px [vertical distance]px [feathering distance]px [shadow color]

Property: **text-align**

Specifies text alignment
This property may take the value(s) below as the property value.

"left"
Align to left when written horizontally (Align to top when written vertically)

"center"
Align to center relative to the line progression direction.

"right"
Align to right when written horizontally (Align to bottom when written vertically)

"justify"
Align on both sides

Property: **text-vertical-align**

Specifies the position of objects in <body> relative to block progression direction.
This property may take the value(s) below as the property value.

"top"
Align to top when written horizontally (Align to right when written vertically) Top alignment (right alignment when vertical text)

"middle"
Align to page center

"bottom"
Align to bottom when written horizontally (Align to left when written vertically)

Property: **align**

Horizontal image alignment. Continuing text does not wrap.
This property may take the value(s) below as the property value.

"left"
Left image alignment (top alignment when vertical text)

"center"
Center image alignment

"right"

Right image alignment (bottom alignment when vertical text)

Property: **vertical-align**

Specifies vertical image alignment relative to the text. This property may take the value(s) below as the property value.

"top"

Top alignment (right alignment when vertical text)

"middle"

Page center alignment

"bottom"

Bottom alignment (left alignment when vertical text)

Property: **linemode**

Specifies text wrapping.

This property may take the value(s) below as the property value.

bool

Property: **float**

Specifies the image's horizontal position and wrapping.

This property may take the value(s) below as the property value.

"left"

Puts the image to the left, continuing text wraps around the right side of the image.

"right"

Puts the image to the right, continuing text wraps around the left side of the image.

“none”:

Neither image alignment nor text wraps.

Property: **letter-spacing**

Specifies the letter spacing.

This property may take the value(s) below as the property value.

[n]px

Specifies the letter spacing as a numeric value. (unit: px)

"maximum"

Displays text with maximum letter spacing.

"big"

Displays text with large letter spacing.

"medium"

Displays text with medium letter spacing.

"small"

Displays text with small letter spacing.

"minimum"

Displays text with minimum letter spacing.

Property: **line-height**

Specifies the line height (vertical spacing).

This property may take the value(s) below as the property value.

[n]px, [n]%

Specifies the line height (vertical spacing) as a numeric value. (unit: px)

"maximum"

Displays text with the maximum line height.

"big"

Displays text with the large line height.

"medium"

Displays text with the medium line height.

"small"

Displays text with the small line height.

"minimum"

Displays text with the minimum line height.

Property: **-ttime-line-height**

Specifies the line spacing.

This property may take the value(s) below as the property value.
[n]px

Property: **column-count**
Column setting. If 0 is specified, "auto (viewer dependent)".
This property may take the value(s) below as the property value.
[n]
0 or larger integer

Property: **fixed-line-char**
Number of characters for one line (fixed). If 0 is specified, "auto (viewer dependent)".
This property may take the value(s) below as the property value.
[n]
0 or larger integer

Properties: **padding-start**
Text block indentation
This property may take the value(s) below as the property value.
[n]px
[n]%,[n]em

Properties: **padding-end**
Indent from the bottom of the entire text block
This property may take the value(s) below as the property value.
[n]px
[n]%,[n]em

Properties: **padding-before**
Indent for the start direction of the text block
This property may take the value(s) below as the property value.
[n]px
[n]%,[n]em

Properties: **padding-after**
Indent for the progression direction of the text block
This property may take the value(s) below as the property value.
[n]px
[n]%,[n]em

Property: **text-indent**
Indent for the first line of text
This property may take the value(s) below as the property value.
[i]em

Property: **margin-before**
The margin for the page start direction (right when vertical text, top when horizontal text)
This property may take the value(s) below as the property value.
[n]px
Specifies as a numeric value. (unit: px)
[n]%
Specifies as a numeric value. (unit: %)
"big"
Displays text with a large margin.
"medium"
Displays text with a medium margin.
"small"
Displays text with a small margin.

Property: **margin-after**
The margin for the page progression direction (left when vertical text, right when horizontal text)

This property may take the value(s) below as the property value.

[n]px

Specifies as a numeric value. (unit: px)

[n]%

Specifies as a numeric value. (unit: %)

"big"

Displays text with a large margin.

"medium"

Displays text with a medium margin.

"small"

Displays text with a small margin.

Property: **margin-start**

The margin for the content start direction (top when vertical text, left when horizontal text)

This property may take the value(s) below as the property value.

[n]px

Specifies as a numeric value. (unit: px)

[n]%

Specifies as a numeric value. (unit: %)

"big"

Displays text with a large margin.

"medium"

Displays text with a medium margin.

"small"

Displays text with a small margin.

Property: **margin-end**

The margin for the content progression direction (bottom when vertical text, right when horizontal text)

This property may take the value(s) below as the property value.

[n]px

Specifies as a numeric value. (unit: px)

[n]%

Specifies as a numeric value. (unit: %)

"big"

Displays text with a large margin.

"medium"

Displays text with a medium margin.

"small"

Displays text with a small margin.

Property: **margin-left**

Left margin (absolute)

This property may take the value(s) below as the property value.

[n]px,[n]%

Specifies the top margin as a numeric value. (unit: px or %)

"big"

Displays text with a large margin.

"medium"

Displays text with a medium margin.

"small"

Displays text with a small margin.

Property: **margin-top**

Top margin (absolute)

This property may take the value(s) below as the property value.

[n]px,[n]%

Specifies the top margin as a numeric value. (unit: px or %)

big

Displays text with a large margin.

medium

Displays text with a medium margin.

small

Displays text with a small margin.

Property: **margin-right**

Right margin (absolute)

This property may take the value(s) below as the property value.

[n]px,[n]%

Specifies the right margin as a numeric value. (unit: px or %)

"big"

Displays text with a large margin.

"medium"

Displays text with a medium margin.

"small"

Displays text with a small margin.

Property: **margin-bottom**

Bottom margin (absolute)

This property may take the value(s) below as the property value.

[n]px,[n]%

Specifies the bottom margin as a numeric value. (unit: px or %)

"big"

Displays text with a large margin.

"medium"

Displays text with a medium margin.

"small"

Displays text with a small margin.

Property: **background-color**

Background color

This property may take the value(s) below as the property value.

colorvalue

Property: **background-color-space**

Specifies the color space for the background color.

This property may take the value(s) below as the property value.

"RGB"

The color space for the background color is RGB.

"SRGB"

The color space for the background color is SRGB.

Property: **background-image**

Specifies a background image.

This property may take the value(s) below as the property value.

url("...")

Property: **background-music**

Specifies background music.

This property may take the value(s) below as the property value.

url("...")

Property: **background-music-loop**

Sets background music looping.

This property may take the value(s) below as the property value.

bool

Properties: **writing-mode**

Direction for writing text

This property may take the value(s) below as the property value.

"vertical-rl"

Specifies vertical text.

"horizontal-tb"

Specifies horizontal text.

"change"

Changes the text direction to the opposite of the default direction.

"default"

Sets the text direction to the direction specified as the default.

Properties: **force-writing-mode**

Sets whether or not to force the writing-mode setting.

This property may take the value(s) below as the property value.

bool

Property: **text-combine-rule**

Specifies the number of characters for horizontal-in-vertical text.

Remarks: In locations where you wish to intentionally use horizontal-in-vertical text, the text-combine property should be clearly specified regardless of the presence of this property.

This property may take the value(s) below as the property value.

[n]

1 or larger integer

Remarks:

When text-combine-rule is specified, the text below is subject to horizontal-in-vertical composition.

For example, when the value for text-combine-rule is 2:

・If the number of continuous alphanumeric characters is less than the value specified by text-combine-rule.

Example:

12 月 The characters "12" are 2 or lower, so they are subject to horizontal-in-vertical composition.

1984 年 The characters "1984" are greater than 2, so they are not subject to horizontal-in-vertical composition.

・Characters with European accent symbols are not subject to horizontal-in-vertical composition.

・Handling half-width text that includes half-width spaces such as sentences in European languages (whether or not to consider a half-width space as a character) is dependent on the text processing system and is not mentioned in this standard. Therefore, you should use the text-combine property when you wish to use horizontal-in-vertical composition with certainty.

Example:

This is a pen.

・When half-width spaces are a portion of the text:	14 total characters, not subject to horizontal-in-vertical composition.
・When half-width spaces are not a portion of the text:	"is" and "a" are subject to horizontal-in-vertical composition.

Property: **text-combine**

Horizontal-in-vertical composition

Remarks: Shall not be specified in em.

This property may take the value(s) below as the property value.

"none"

Nothing

"horizontal"

Performs horizontal-in-vertical composition.

Property: **window-type**

Screen direction to recommend

This property may take the value(s) below as the property value.

"portrait"

Recommends portrait.

"landscape"

Recommends landscape.

Property: **force-window-type**

Sets whether or not to force window-type.

This property may take the value(s) below as the property value.

bool

Property: **first-letter**

A pseudo element that denotes the first character. Use for specifying dropcaps.

Remarks: The *p tag has been deprecated. Styles that attempt to be written to the p tag are written to the body selector.

Property: **text-offset**

Offset of the display position. The relative offset from the value that is valid at that time.

Remarks: When a <div>/<span> appears that has the text-offset property or a separate exclusive T-Time setting <offset/> tag appears, be aware that the offset becomes the relative value from there without clearing the value.

This property may take the value(s) below as the property value.

[horizontal distance]px [vertical distance]px

Property: **from-page-end**

Displays n lines from the end of the text field.

This property may take the value(s) below as the property value.

[n](number of lines)

The number of lines to display. (unit: lines)

Property: **line-break**

Sets Japanese line breaking rules and the degree of line breaking. Specify the line head wrap characters with top-prohibit-char, specify the line end wrap characters with end-prohibit-char.

This property may take the value(s) below as the property value.

"auto"

Display system dependent line breaking rules

"newspaper"

Relaxed line breaking rules for text with short lines like newspapers

"normal"

Apply normal line breaking rules.

"strict"

Apply strict line breaking rules.

"keep-all"

Do not break lines in words.

"none"

No line breaking rules

"run-down"

Wrap-to-next rules

Remarks: Changed from β of "run_down".

Property: **hanging-punctuation**

Specifies hanging punctuation processing. Hanging characters are specified with hanging-char.

This property may take the value(s) below as the property value.

"first"

Characters hang at the start of the line.

"last"

Characters hang at the end of the line.

"allow-end"

Has hanging punctuation processing.

"force-end"

If the end character is a target character, adjust the character spacing so it hangs even if it fits within the frame.

"none"

No hanging punctuation processing

Remarks: Requires a check to determine if this value is required.

Properties: **hanging-char**

Registers hanging characters.
This property may take the value(s) below as the property value.
Hanging characters

Property: **top-prohibit-char**

Registers line head wrap characters.
This property may take the value(s) below as the property value.
Line head wrap characters

Property: **end-prohibit-char**

Registers line end wrap characters.
This property may take the value(s) below as the property value.
Line end wrap characters

Property: **ruby-flag**

Sets whether or not to display ruby characters.
This property may take the value(s) below as the property value.
bool

Property: **force-ruby-setting**

Sets whether or not to force the ruby character display setting specified by the ruby-flag property.
This property may take the value(s) below as the property value.
bool

Property: **running-head-visible**

Sets whether or not to display the running head.
This property may take the value(s) below as the property value.
bool

Property: **running-head-auto-hide**

Automatically hides the running head on pages with a heading.
This property may take the value(s) below as the property value.
bool

Property: **running-head-align**

Running head position alignment
This property may take the value(s) below as the property value.
"left"
Displays the running head in the upper left.
"center"
Displays the running head in the center.
"right"
Displays the running head in the upper right.

Property: **running-head-font-family**

Specifies the running head font family. If multiple font names are specified, the first item has precedence.
This property may take the value(s) below as the property value.
Font name

Property: **running-head-font-size**

Running head text size
This property may take the value(s) below as the property value.
[n]px

Property: **running-head-size-lock**

Locks the running head text size
This property may take the value(s) below as the property value.
bool

Property: **running-head-color**

Running head text color

This property may take the value(s) below as the property value.

colorvalue

Property: **running-head-shadow**

Applies a shadow to the running head display style.

This property may take the value(s) below as the property value.

[horizontal distance]px [vertical distance]px [feathering distance]px [shadow color]

Property: **link-color**

Specifies the book note color.

This property may take the value(s) below as the property value.

colorvalue

Property: **link-color-space**

Specifies the color space for the book note color.

This property may take the value(s) below as the property value.

"RGB"

The color space for the book note color is RGB.

"SRGB"

The color space for the book note color is SRGB.

Property: **visible**

Display heading

This property may take the value(s) below as the property value.

bool

Property: **heading-font-family**

Specifies the heading text font.

This property may take the value(s) below as the property value.

Font name

Properties: **heading-font-size**

Specifies the heading text size.

This property may take the value(s) below as the property value.

[n]px

Properties: **height**

Horizontal line thickness

This property may take the value(s) below as the property value.

[n]px

[n]em

Properties: **width**

Horizontal line width

This property may take the value(s) below as the property value.

[n]px

[n]em, [n]%

Properties: **headspace**

Compresses full-width spaces for the head of line during PDA mode (screen size less than 320x320).

This property may take the value(s) below as the property value.

"compress"

Always automatically compress.

"normal"

Never automatically compress.

"auto"

Automatically compress in PDA mode, do not automatically compress in PC mode.

Property: **indentspace**

Compresses the head of line indentation/indentation amount during PDA mode.

This property may take the value(s) below as the property value.

"compress"

Always automatically compress.

"normal"

Never automatically compress.

"auto"

Automatically compress in PDA mode, do not automatically compress in PC mode.

Property: **linespacing**

Compresses line spacing during PDA mode.

This property may take the value(s) below as the property value.

"compress"

Always automatically compress.

"normal"

Never automatically compress.

"auto"

Automatically compress in PDA mode, do not automatically compress in PC mode.

Property: **tabspacing**

Compresses head of line tab width during PDA mode.

This property may take the value(s) below as the property value.

"compress"

Always automatically compress.

"normal"

Never automatically compress.

"auto"

Automatically compress in PDA mode, do not automatically compress in PC mode.

C.4.3 Property-selector combinations

The combinations of the properties and corresponding selectors are given in Table C.5.

Table C.5 – Property-selector correnspondence

		Type	other	block			inline	
		Selector name	body	div	h1,h2…h9	hr	span	em
	Property	Property value						
Font/character embellishments	**font-family**							
		Font name	✓	✓	✓		✓	✓
	font-style							
		"normal"		✓	✓		✓	✓
		"italic"		✓	✓		✓	✓
		"oblique"		✓	✓		✓	✓
	font-variant							
		"normal"		✓	✓		✓	✓
		"small-caps"		✓	✓		✓	✓
	font-size							
		[n]px	✓	✓	✓		✓	✓
		[n]%, [n]em	✓	✓	✓		✓	✓
		"maximum"	✓	✓	✓		✓	✓
		"big"	✓	✓	✓		✓	✓
		"medium"	✓	✓	✓		✓	✓
		"small"	✓	✓	✓		✓	✓
		"minimum"	✓	✓	✓		✓	✓
	font-base							
		"last"		✓			✓	
		"default"		✓			✓	
	color							
		colorvalue	✓	✓	✓		✓	✓
	color-space							
		"RGB"	✓	✓	✓		✓	✓
		"SRGB"	✓	✓	✓		✓	✓
	filter							
		"invert()"		✓			✓	✓

		Type	other	block			inline	
		Selector name	body	div	h1,h2…h9	hr	span	em
	Property	Property value						
Font/character embellishments	**font-weight**							
		"normal"	✓	✓	✓		✓	✓
		"bold"	✓	✓	✓		✓	✓
		"bolder"	✓	✓	✓		✓	✓
		"lighter"	✓	✓	✓		✓	✓
		100	✓	✓	✓		✓	✓
		200	✓	✓	✓		✓	✓
		300	✓	✓	✓		✓	✓
		400	✓	✓	✓		✓	✓
		500	✓	✓	✓		✓	✓
		600	✓	✓	✓		✓	✓
		700	✓	✓	✓		✓	✓
		800	✓	✓	✓		✓	✓
		900	✓	✓	✓		✓	✓
		"inherit"	✓	✓	✓		✓	✓
	text-decoration							
		"underline"		✓	✓		✓	✓
		"none"		✓	✓		✓	✓
	text-emphasis-style							
		"none"		✓			✓	✓
		"filled sesame"		✓			✓	✓
		"filled double-circle"		✓			✓	✓
		"filled dot"		✓			✓	✓
		"filled circle"		✓			✓	✓
		"filled triangle"		✓			✓	✓
		"open sesame"		✓			✓	✓
		"open double-circle"		✓			✓	✓
		"open dot"		✓			✓	✓
		"open circle"		✓			✓	✓
		"open triangle"		✓			✓	✓
		Text		✓			✓	✓
	text-shadow							
		[horizontal distance]px [vertical distance]px [feathering distance]px [shadow color]	✓				✓	✓

		Type	other	block			inline	
		Selector name	body	div	h1,h2…h9	hr	span	em
	Property	Property value						
Alignment/wrapping	**text-align**							
		"left"	✓	✓		✓		
		"center"	✓	✓		✓		
		"right"	✓	✓		✓		
		"justify"	✓	✓		✓		
	text-vertical-align							
		"top"	✓					
		"middle"	✓					
		"bottom"	✓					
	align							
		"left"						
		"center"						
		"right"						
	vertical-align							
		"top"						
		"middle"						
		"bottom"						
	linemode							
		bool	✓	✓				
	float							
		"left"						
		"right"						
		"none"						
Character spacing/line spacing/columns/number of characters	**letter-spacing**							
		[n]px	✓	✓	✓		✓	✓
		"maximum"	✓	✓	✓		✓	✓
		"big"	✓	✓	✓		✓	✓
		"medium"	✓	✓	✓		✓	✓
		"small"	✓	✓	✓		✓	✓
		"minimum"	✓	✓	✓		✓	✓
	line-height							
		[n]px, [n]%	✓	✓	✓			
		"maximum"	✓	✓	✓			
		"big"	✓	✓	✓			
		"medium"	✓	✓	✓			
		"small"	✓	✓	✓			
		"minimum"	✓	✓	✓			
	-ttime-line-height							
		[n]px					✓	

		Type	other	block			inline	
		Selector name	body	div	h1,h2…h9	hr	span	em
	Property	Property value						
columns/ number of characters	**column-count**							
		[n]	✓					
	fixed-line-char							
		[n]	✓					
Indentation	**padding-start**							
		[n]px		✓				
		[n]%,[n]em		✓				
	padding-end							
		[n]px		✓				
		[n]%,[n]em		✓				
	padding-before							
		[n]px		✓				
		[n]%,[n]em		✓				
	padding-after							
		[n]px		✓				
		[n]%,[n]em		✓				
	text-indent							
		[i]em		✓				
Margins	**margin-before**							
		[n]px	✓	✓	✓	✓		
		[n]%	✓					
		"big"	✓					
		"medium"	✓					
		"small"	✓					
	margin-after							
		[n]px	✓	✓	✓	✓		
		[n]%	✓					
		"big"	✓					
		"medium"	✓					
		"small"	✓					
	margin-start							
		[n]px	✓	✓	✓	✓		
		[n]%	✓					
		"big"	✓					
		"medium"	✓					
		"small"	✓					

		Type	other	block			inline	
		Selector name	body	div	h1,h2...h9	hr	span	em
	Property	Property value						
Margins	**margin-end**							
		[n]px	✓	✓	✓	✓		
		[n]%	✓					
		"big"	✓					
		"medium"	✓					
		"small"	✓					
	margin-left							
		[n]px,[n]%	✓					
		"big"	✓					
		"medium"	✓					
		"small"	✓					
	margin-top							
		[n]px,[n]%	✓					
		big	✓					
		medium	✓					
		small	✓					
	margin-right							
		[n]px,[n]%	✓					
		"big"	✓					
		"medium"	✓					
		"small"	✓					
	margin-bottom							
		[n]px,[n]%	✓					
		"big"	✓					
		"medium"	✓					
		"small"	✓					
Background	**background-color**							
		colorvalue	✓					
	background-color-space							
		"RGB"	✓					
		"SRGB"	✓					
	background-image							
		url("・・・")	✓					
	background-music							
		url("・・・")	✓					
	background-music-loop							
		bool	✓					

		Type	other	block			inline	
		Selector name	body	div	h1,h2…h9	hr	span	em
	Property	Property value						
Text direction	**writing-mode**							
		"vertical-rl"	✓					
		"horizontal-tb"	✓					
		"change"	✓					
		"default"	✓					
	force-writing-mode							
		bool	✓					
Horizontal-in-vertical	**text-combine-rule**							
		[n]	✓					
	text-combine							
		"none"					✓	
		"horizontal"					✓	
Screen orientation	**window-type**							
		"portrait"	✓					
		"landscape"	✓					
	force-window-type							
		bool	✓					
Other formatting	**first-letter**		✓					
	text-offset							
		[horizontal distance]px [vertical distance]px		✓			✓	
	from-page-end							
		[n] (number of lines)		✓				
Line breaking rules	**line-break**							
		"auto"	✓					
		"newspaper "	✓					
		"normal"	✓					
		"strict"	✓					
		"keep-all"	✓					
		"none"	✓					
		"run-down"	✓					
	hanging-punctuation							
		"first"	✓					
		"last"	✓					
		"allow-end"	✓					
		"force-end"	✓					
		"none"	✓					

		Type	other	block			inline	
		Selector name	body	div	h1,h2…h9	hr	span	em
	Property	Property value						
Line breaking rules	**hanging-char**							
		Hanging characters	✓					
	top-prohibit-char							
		Line head wrap characters	✓					
	end-prohibit-char							
		Line end wrap characters	✓					
Ruby characters	**ruby-flag**							
		bool	✓					
	force-ruby-setting							
		bool	✓					
Running head	**running-head-visible**							
		bool	✓					
	running-head-auto-hide							
		bool	✓					
	running-head-align							
		"left"	✓					
		"center"	✓					
		"right"	✓					
	running-head-font-family							
		Font name	✓					
	running-head-font-size							
		[n]px	✓					
	running-head-size-lock							
		bool	✓					
	running-head-color							
		colorvalue	✓					
	running-head-shadow							
		[horizontal distance]px [vertical distance]px [feathering distance]px [shadow color]	✓					

		Type	other	block			inline	
		Selector name	body	div	h1,h2...h9	hr	span	em
	Property	Property value						
Link	**link-color**							
		colorvalue	✓					
	link-color-space							
		"RGB"	✓					
		"SRGB"	✓					
Headings	**visible**							
		bool			✓			
	heading-font-family							
		Font name	✓					
	heading-font-size							
		[n]px	✓					
Size	**height**							
		[n]px				✓		
		[n]em				✓		
	width							
		[n]px				✓		
		[n]em, [n]%				✓		
PDA mode	**headspace**							
		"compress"	✓					
		"normal"	✓					
		"auto"	✓					
	indentspace							
		"compress"	✓					
		"normal"	✓					
		"auto"	✓					
	linespacing							
		"compress"	✓					
		"normal"	✓					
		"auto"	✓					
	tabspacing							
		"compress"	✓					
		"normal"	✓					
		"auto"	✓					

		Type	inline			Others		Pseudo-classes
		Selector name	a	img	mask	offset	video	a:link, a:hover, a:active, a:visited
	Property	Property value						
Font/character embellishments	**color**							
		colorvalue	✓		✓			✓
	color-space							
		"RGB"	✓		✓			✓
		"SRGB"	✓		✓			✓
	text-decoration							
		"underline"	✓					
		"none"	✓					
Alignment/wrapping	**align**							
		"left"		✓			✓	
		"center"		✓			✓	
		"right"		✓			✓	
	vertical-align							
		"top"		✓			✓	
		"middle"		✓			✓	
		"bottom"		✓			✓	
	linemode							
		bool						
	float							
		"left"		✓			✓	
		"right"		✓			✓	
		"none"		✓			✓	
Other formatting	**text-offset**							
		[horizontal distance]px [vertical distance]px				✓		

C.5 Comments to specifications

C.5.1 Block tag behavior

Tags listed in Table C.6 make the portion enclosed by the start and end tags an independent block.

Table C.6 – Tags that make independent blocks

Tag	Notes
<div>	
<h1>	
<h2>	
<h3>	
<h4>	
<h5>	
<h6>	
<h7>	
<h8>	
<h9>	
<hr>	The drawn line is subject
<marquee>	
<tr>	The text included in the child elements <th> and <td> is subject

C.5.2 Style data parsing

Style data parsing when switching between vertical text and horizontal text is defined as listed below.

For border-*, margin-*, padding-*

- top/left/bottom/right: Parsing does not change for vertical text and horizontal text.
- start/before/after/end: Parsing changes for vertical text and horizontal text and becomes as shown in Table C.7.

Table C.7 – Property parsing

Property	Meaning	Parsing during horizontal text	Parsing during vertical text
start	Line start direction	left	top
end	Line progression direction	right	bottom
before	Block start direction	top	right
after	Block progression direction	bottom	left

For "line start direction/progression direction" and "block start direction/progression direction" in Table C.7, refer to Figure C.6.

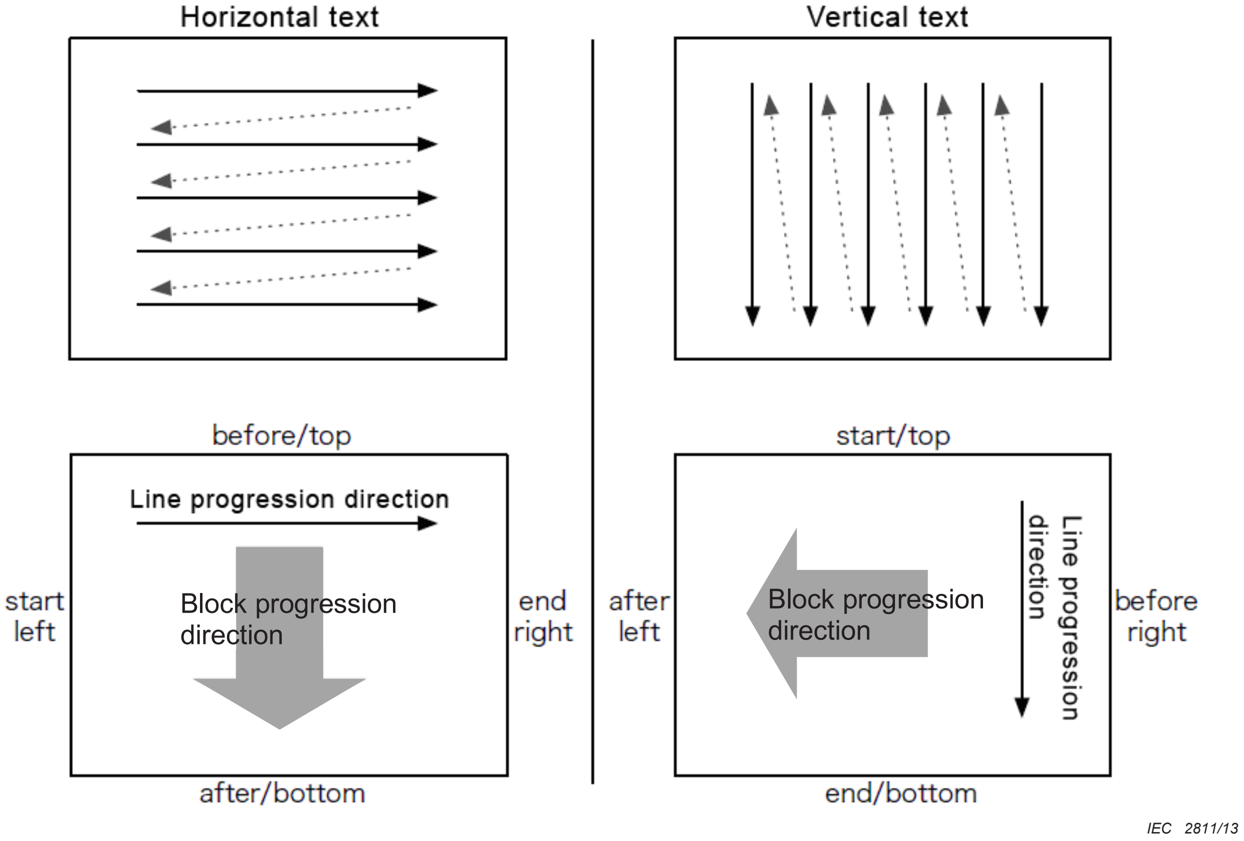
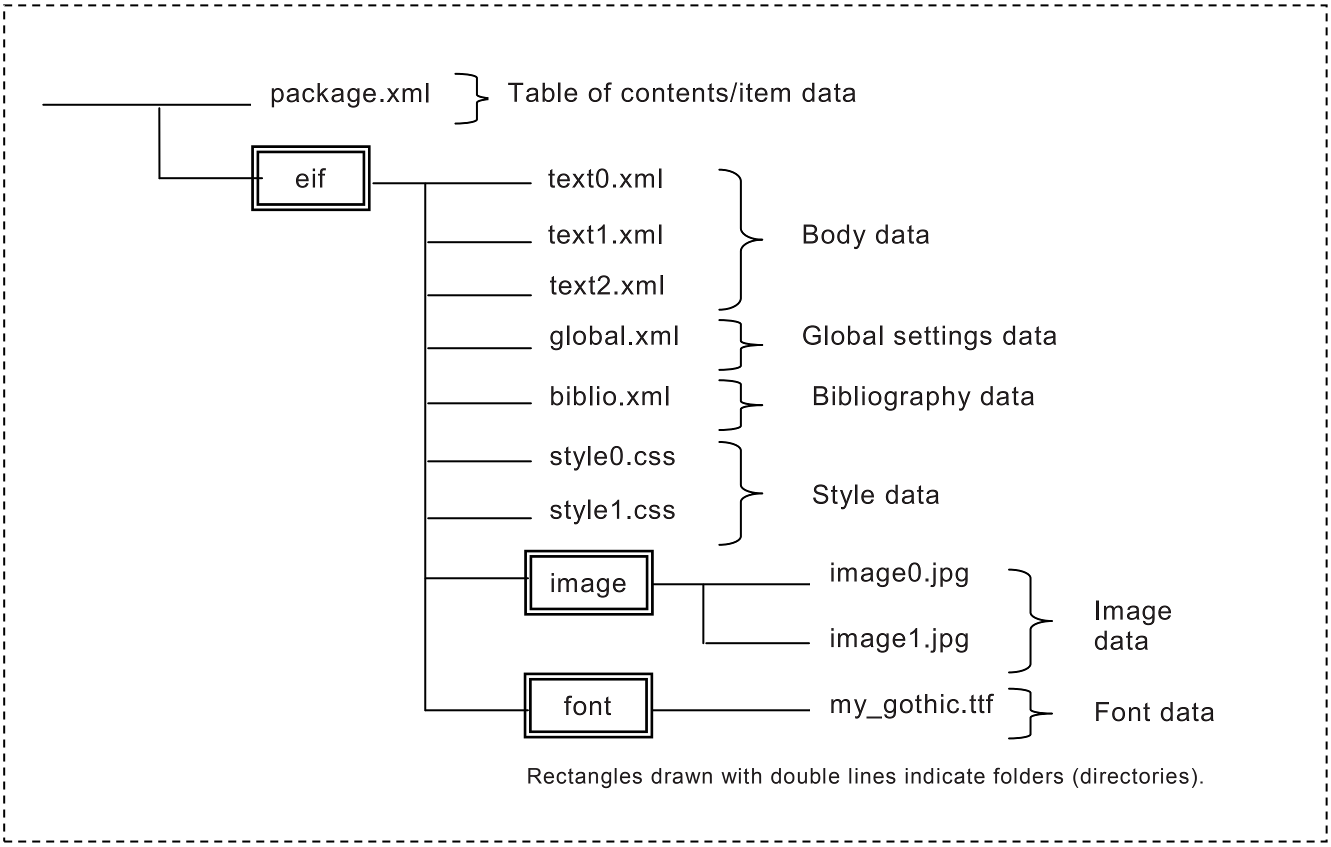


Figure C.6 – Content/page progression directions

C.5.3 Folder structure

The following statements apply to the folder structure.

- The folder (directory) structure in ESP format specification is not specifically defined.
- An example of the folder structure is shown in Figure C.7. (Note that file names other than package.xml are mere examples.)
- You should place the files in an independent folder (directory) for each item of content, and you should not place files that are not structural elements of the content in that folder (directory).



IEC 2812/13

Figure C.7 – Folder (directory) structure example

C.5.4 Indicating vendor-specific information

C.5.4.1 Using < proprietary> tag

Considering the needs to function as a “hub” in converting from one format to another, ESP format provides several methods to record vendor-specific functions, i.e. functions of contents that are not to be reflected in the destination format but should be recorded in the ESP (hub) format.

The <proprietary> tag is one such method available in ESP format. The <proprietary> tag may be written in the global settings data (<global_setting>) or in a child element of <local_setting> in the body data. It has the following attributes.

NOTE However, whether or not the supported processing is performed is dependent on the tool.

Code example:

```
<proprietary vendor=”ttime”>
<production_rule v3 = “true”/>
</proprietary>
```

When ESP format has instrinsic methods (tags/attributes/style data) available to represent the functions, it is recommended such instrinsic methods should be used rather than <proprietary> tag.

C.5.4.2 Using namespaces in <head> tag

ESP format allows using any tags with namespace prefix in <head> tag as long as the namespace prefix is resolvable.

C.5.5 Minimum/small/medium/large/maximum values that indicate text size, text spacing, and line spacing

This type of code in the existing format is viewer dependent and the actual rendering may differ, but it has been left in the style data code for ESP format because converting it to numeric values does not always express the intentions of the creator.

Table C.8 shows an example of converting this code to numeric values.

NOTE This conversion to numeric values is just an example. The creators did not necessarily create their content with these numeric values in mind, so use them as a reference when converting to other formats.

Table C.8 – An example of conversion to numeric values

Text size	Corresponding numeric value
minimum	60%
small	80%
medium	100%
large	120%
maximum	140%

C.5.6 Font size descriptions

The following precautions are required for font size descriptions.

– A font size of 0 or lower shall not be used.

– In the specification related to font size, locations with [n] are interpreted as an integer of 1 or higher for font size.

  NOTE In all other locations, [n] is interpreted as an integer of 0 or higher unless specifically noted.

– If the unit is em, decimal values may also be used.

– However, do not write these values in a manner that omits the integer portion, such as ".5".

C.6 Color names

When specifying a color, the color name listed in Table C.9 can be used in addition to the color code. The color names are case-insensitive. When using a color name, check that it matches the color code specified in the "Color code" column.

Table C.9 – Color names

Color name	Color code (hex triplet)
black	#000000
silver	#C0C0C0
gray	#808080
white	#FFFFFF
maroon	#800000
red	#FF0000
purple	#800080
fuchsia	#FF00FF
green	#008000
lime	#00FF00
olive	#808000
yellow	#FFFF00
navy	#000080
blue	#0000FF
teal	#008080
aqua	#00FFFF

C.7 Specification of ESP format in the RELAX NG compact syntax

```
default namespace a = "http://ebformat.jp"

namespace empty = ""

bool = "true" | "false"

object-tags = img | audio | video

char-or-external-char = text | external_char

char-or-external-char-or-ruby = text | external_char | ruby

block-tags =

  \div

  | h1

  | h2

  | h3

  | h4

  | h5

  | h6
  | h7

  | h8

  | h9

  | marquee

  | table

  | action

  | tts

  | section

  | page_break

  | reghead

  | hr

inline-tags-core =

  br

  | em

  | span

  | external_char

  | mlg

  | sub

  | sup

  | phead

  | ptail

  | map

  | ruby

  | case

  | nocase

  | mask

  | column_break
```

```
inline-tags-no-anchor = inline-tags-core | object-tags | offset

inline-tags = inline-tags-core | object-tags | a | offset

text-tags = block-tags | inline-tags

turning-page-val = "on" | "off" | "forward" | "back"

turning-page-control-val =

  "on" | "off" | "forward" | "back" | "lastpage"

body =

  element body {

    body-attlist,

    (text

     | text-tags

     | search_table

     | comic_cell_type_body

     | comic_page_type_body

     | dict_item

     | ref

     | split

     | column

     | search_page)*

  }

body-attlist &=

  attribute class { text }?,

  [ a:defaultValue = "text" ]

  attribute type { "text" | "search" | "comic" | "dict" }?,

  attribute turning_page_control { turning-page-control-val }?


start |= head
```

```
head =
  element head {
    head-attlist,
    (title
     | link
     | pdef
     | local_setting
     | anyElement_head
    )*
  }
head-attlist &= empty
html = element html { html-attlist, head, body }
html-attlist &=
           attribute xml:lang { text }?

anyElement_head = element * – (a:*|empty:*) {
           attribute * {text}*,
           (text
         | anyElement_head
        )*
           }*


page_break = element page_break { page_break-attlist, empty }
page_break-attlist &=
  attribute method { "normal" | "odd" | "even" | "left" | "right" }?,
  attribute turning_page_control { turning-page-control-val }?
```

```
column_break = element column_break { column_break-attlist, empty }

column_break-attlist &= empty

br = element br { br-attlist, empty }

br-attlist &=

  attribute class { text }?,

  attribute clear { "all" | "left" | "right" }?

\div = element div { div-attlist, (text | text-tags)* }

div-attlist &=

  attribute class { text }?,

  attribute id { text }?,

  attribute level { text }?

em = element em { em-attlist, (text | inline-tags)* }

em-attlist &= attribute class { text }?

h1 = element h1 { h1-attlist, (text | inline-tags | \div)* }

h1-attlist &=

  attribute class { text }?,

  attribute caption { text }?

h2 = element h2 { h2-attlist, (text | inline-tags | \div)* }

h2-attlist &=

  attribute class { text }?,

  attribute caption { text }?

h3 = element h3 { h3-attlist, (text | inline-tags | \div)* }

h3-attlist &=

  attribute class { text }?,

  attribute caption { text }?

h4 = element h4 { h4-attlist, (text | inline-tags | \div)* }

h4-attlist &=
  attribute class { text }?,

  attribute caption { text }?

h5 = element h5 { h5-attlist, (text | inline-tags | \div)* }

h5-attlist &=

  attribute class { text }?,

  attribute caption { text }?

h6 = element h6 { h6-attlist, (text | inline-tags | \div)* }

h6-attlist &=

  attribute class { text }?,

  attribute caption { text }?

h7 = element h7 { h7-attlist, (text | inline-tags | \div)* }

h7-attlist &=

  attribute class { text }?,

  attribute caption { text }?

h8 = element h8 { h8-attlist, (text | inline-tags | \div)* }

h8-attlist &=

  attribute class { text }?,

  attribute caption { text }?

h9 = element h9 { h9-attlist, (text | inline-tags | \div)* }

h9-attlist &=

  attribute class { text }?,

  attribute caption { text }?

span = element span { span-attlist, (text | inline-tags)* }

span-attlist &=

  attribute class { text }?,

  attribute id { text }?,

  attribute type { text }?,
  attribute lang_code { text }?,
  attribute xml:lang { text }?
offset = element offset { offset-attlist, empty }
offset-attlist &= attribute class { text }?
external_char = element external_char { external_char-attlist, (span | img)? }
external_char-attlist &=
  attribute system { text }?,
  attribute code { text }?,
  attribute glyph_data { text }?,
  attribute priority { text }?,
  attribute alt { text }?
mlg = element mlg { mlg-attlist, (text | inline-tags)* }
mlg-attlist &= empty
a = element a { a-attlist, (text | inline-tags-no-anchor)* }
a-attlist &=
  attribute class { text }?,
  attribute id { text }?,
  attribute href { text }?,
  attribute target { text }?,
  attribute ref_itemid { text }?,
  attribute title { text }?,
  attribute ref_id { text }?,
  attribute shake { bool }?,
  attribute background { bool }?,
  attribute cmd { text }?
hr = element hr { hr-attlist, empty }
hr-attlist &= attribute class { text }?
```

```
sub = element sub { sub-attlist, (text | inline-tags)* }

sub-attlist &= empty

sup = element sup { sup-attlist, (text | inline-tags)* }

sup-attlist &= empty

marquee =

  element marquee {

    marquee-attlist, (text | external_char | ruby | object-tags | span)*

  }

marquee-attlist &= empty

table = element table { table-attlist, tr+ }

table-attlist &= empty

tr = element tr { tr-attlist, (th | td)+ }

tr-attlist &= empty

th = element th { th-attlist, (text | inline-tags)* }

th-attlist &=

          attribute rowspan { text }?,

          attribute colspan { text }?

td = element td { td-attlist, (text | inline-tags)* }

td-attlist &=

          attribute rowspan { text }?,

          attribute colspan { text }?

img = element img { img-attlist, empty }

img-attlist &=

  attribute class { text }?,

  attribute id { text }?,

  attribute src { text },

  attribute alt { text }?,
  attribute longdesc { text }?,
  attribute width { text }?,
  attribute height { text }?,
  attribute usemap { text }?,
  attribute line { text }?,
  attribute scale { text }?,
  attribute bordercolor { text }?,
  attribute border { text }?,
  attribute hspace { text }?,
  attribute vspace { text }?,
  attribute dspace { text }?,
  attribute snap { "frame" }?,
  attribute linemode { bool }?,
  attribute targetplane { "text" }?,
  attribute drawing { "tile" }?,
  attribute color { text }?,
  attribute opacity { text }?,
  attribute loop { text }?,
  attribute start { "auto" | "event" }?,
  attribute a { text }?,
  attribute x { text }?,
  attribute y { text }?,
  attribute o { bool }?,
  attribute mode { text }?,
  attribute shrink { "screen" }?,
  attribute startpoint { text }?,
  attribute preview { bool }?
```

```
pdef = element pdef { pdef-attlist, empty }

pdef-attlist &=

  attribute src { text },

  attribute alt { text }?,

  attribute id { text },

  attribute width { text }?,

  attribute height { text }?,

  attribute usemap { text }?,

  attribute bordercolor { text }?,

  attribute border { text }?,

  attribute hspace { text }?,

  attribute vspace { text }?,

  attribute color { text }?,

  attribute opacity { text }?,

  attribute a { text }?,

  attribute x { text }?,

  attribute y { text }?,

  attribute o { bool }?,

  attribute mode { text }?,

  attribute shrink { "screen" }?,

  attribute startpoint { text }?,

  attribute preview { bool }?

phead = element phead { phead-attlist, empty }

phead-attlist &=

  attribute ref_id { text },

  attribute delay { text }?

ptail = element ptail { ptail-attlist, empty }
```

```
ptail-attlist &=

  attribute ref_id { text },

  attribute delay { text }?

area = element area { area-attlist, empty }

area-attlist &=

  attribute alt { text }?,

  [ a:defaultValue = "poly" ]

  attribute shape { "rect" | "circle" | "poly" | "default" }?,

  attribute coords { text }?,

  attribute href { text }?,

  attribute target { text }?,

  attribute title { text }?,

  attribute ref_itemid { text }?,

  attribute ref_id { text }?,

  attribute shake { bool }?,

  attribute background { bool }?,

  attribute cmd { text }?

map = element map { map-attlist, (block-tags | area)+ }

map-attlist &=

  attribute id { text }?,

  attribute name { text }?,

  attribute title { text }?

link = element link { link-attlist, empty }

link-attlist &=

  attribute rel { "stylesheet" | "alternate_stylesheet" }?,

  attribute href { text }?,

  attribute type { text }?,
  attribute media { text }?

ruby =

  element ruby {

    ruby-attlist,

    ((rb, rt) | (rb, rp, rt, rp) | (rbc, rtc) | (rbc, rp, rtc, rp))

  }

ruby-attlist &= empty

rb = element rb { rb-attlist, (text | inline-tags)* }

rb-attlist &= empty

rt = element rt { rt-attlist, (text | inline-tags)* }

rt-attlist &= empty

rbc = element rbc { rbc-attlist, rb+ }

rbc-attlist &= empty

rtc = element rtc { rtc-attlist, rt+ }

rtc-attlist &= empty

rp = element rp { rp-attlist, (text | external_char)* }

rp-attlist &= empty

audio = element audio { audio-attlist, empty }

audio-attlist &= attribute src { text }?

video = element video { video-attlist, empty }

video-attlist &= attribute src { text }?

action = element action { action-attlist, empty }

action-attlist &=

  attribute onopen { text }?,

  attribute onclose { text }?,

  attribute background { bool }?,

  attribute cmd { text }?
```

```
case = element case { case-attlist, (text | text-tags)* }

case-attlist &=

  attribute dir { "vertical" | "horizontal" }?,

  attribute cpu { text }?,

  attribute os { text }?,

  attribute screen { "pda" | "portrait" | "landscape" }?

nocase = element nocase { nocase-attlist, (text | text-tags)* }

nocase-attlist &=

  attribute dir { "vertical" | "horizontal" }?,

  attribute cpu { text }?,

  attribute os { text }?,

  attribute screen { "pda" | "portrait" | "landscape" }?

mask = element mask { mask-attlist, (text | inline-tags)* }

mask-attlist &=

  attribute class { text }?,

  attribute initial_flag { "on" | "off" }?,

  attribute trigger_id { text }?,

  attribute id { text }?,

  attribute mask_type { "default" | "color" }?,

  attribute hold_flag { "scope" | "on_power" | "save" }?

tts = element tts { tts-attlist, (text | text-tags)* }

tts-attlist &= attribute speaker { "male" | "female" | "child" }?

permission =

  element permission { permission-attlist, (text | text-tags)* }

permission-attlist &=

  attribute type {

    "print"
    | "copy"

    | "reading"

    | "export"

    | "download"

    | "irda"

    | "reading_on_ppc"

    | "conv_to_palm"

  },

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

section = element section { section-attlist, (text | text-tags)* }

section-attlist &= attribute part { "toc" | "cover" }?

inline-tags-and-p = inline-tags

dict-tags = inline-tags | reghead | subhead

ref = element ref { ref-attlist, char-or-external-char* }

ref-attlist &=

  attribute id { text },

  attribute type { text }?

dict_item =

  element dict_item {

    dict_item-attlist,

    (text

     | dict-tags

     | gender

     | psp

     | glabel

     | pronunciation

     | inflec
    | lang

    | slabel

    | spellout

    | variant

    | etymology)*

  }

dict_item-attlist &=

  attribute type { text }?,

  attribute id { text },

  attribute rank { text }?,

  attribute level { text }?,

  attribute page_break { text }?,

  attribute turning_page_control { turning-page-val }?,

  attribute revision { text }?,

  attribute delete { bool }?

reghead = element reghead { reghead-attlist, headword+, key* }

reghead-attlist &= empty

headword = element headword { headword-attlist, char-or-external-char* }

headword-attlist &=

  attribute type { text }?,

  attribute table_id { text },

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

key = element key { key-attlist, text }

key-attlist &= attribute type { text }?

meaning = element meaning { meaning-attlist, inline-tags-and-p* }

meaning-attlist &=

  attribute type { text }?,
  attribute subid { text }?,

  attribute level { text }?,

  attribute no { text }?

example = element example { example-attlist, inline-tags-and-p* }

example-attlist &= attribute type { text }?

subhead =

  element subhead {

    subhead-attlist, subheadword+, meaning*, example*, key*

  }

subhead-attlist &=

  attribute type { text }?,

  attribute subid { text }?

subheadword =

  element subheadword { subheadword-attlist, char-or-external-char* }

subheadword-attlist &=

  attribute type { text }?,

  attribute subid { text }?

split = element split { split-attlist, inline-tags-and-p* }

split-attlist &= attribute level { text }?

column = element column { column-attlist, inline-tags-and-p* }

column-attlist &=

  attribute type { text }?,

  attribute subid { text }

gender = element gender { gender-attlist, inline-tags* }

gender-attlist &= empty

glabel = element glabel { glabel-attlist, inline-tags* }

glabel-attlist &= empty
```

```
pronunciation =
  element pronunciation { pronunciation-attlist, inline-tags* }
pronunciation-attlist &=
  [ a:defaultValue = "IPA" ] attribute phonetic_notation { text }?
psp = element psp { psp-attlist, inline-tags* }
psp-attlist &= empty
inflec = element inflec { inflec-attlist, inline-tags* }
inflec-attlist &= empty
lang = element lang { lang-attlist, inline-tags* }
lang-attlist &= empty
slabel = element slabel { slabel-attlist, inline-tags* }
slabel-attlist &= empty
spellout = element spellout { spellout-attlist, inline-tags* }
spellout-attlist &= attribute org { text }?
variant = element variant { variant-attlist, inline-tags* }
variant-attlist &= empty
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, char-or-external-char*

  }

search_page_title-attlist &= 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_only" | "matches_first" }?

key_input_region_prompt =

  element key_input_region_prompt {

    key_input_region_prompt-attlist, char-or-external-char*

  }

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 id { text }

search_link_title =

  element search_link_title {

    search_link_title-attlist, char-or-external-char*

  }

search_link_title-attlist &=

  attribute xml:lang { text }?,
  attribute lang_code { text }?
etymology = element etymology { etymology-attlist, inline-tags* }
etymology-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_space { "RGB" }?,
  attribute bg_color { text }?
comic_page_type_body =
  element comic_page_type_body { comic_page_type_body-attlist, page+ }
comic_page_type_body-attlist &=
  attribute bg_color_space { "RGB" }?,
  attribute bg_color { text }?,
  attribute order { "RT-LB" | "LT-RB" | "RB-LT" | "LB-RT" }?
cell = element cell { cell-attlist, (cell_scene | speech | map)* }
cell-attlist &=
  attribute src { text },
  attribute id { text }?,
  attribute bg_color { text }?,
  attribute bg_color_space { "RGB" }?,
  [ a:defaultValue = "false" ] attribute fit { bool }?,
  attribute effect {
    "fade-in"
    | "dissolve"
    | "slide-right"
    | "slide-left"
    | "slide-up"
    | "slide-down"
    | "overwrite-right"
    | "overwrite-left"
    | "overwrite-up"
    | "overwrite-down"
    | "box-rightdown"
    | "box-leftdown"
    | "box-rightup"
    | "box-leftup"
    | "box-center"
    | "box-free"
    | "pushout-right"
    | "pushout-left"
    | "pushout-up"
    | "pushout-down"
    | "wipe-right"
    | "wipe-left"
    | "wipe-up"
    | "wipe-down"
    | "wipe-vertical-in"
    | "wipe-vertical-out"
    | "wipe-horizontal-in"
    | "wipe-horizontal-out"
    | "wipe-center-in"
    | "wipe-center-out"
    | "randomblock"
  }?,
  attribute effect_time { text }?,
  attribute effect_pos { text }?,
  attribute scroll {
    "custom" | "up" | "down" | "left" | "right" | "no"
  }?,
  attribute scroll_time { text }?,
  attribute vibration { bool }?,
  attribute display_vibration { bool }?,
  attribute vibration_direction { "vertical" | "horizontal" }?,
  attribute vibration_time { text }?,
  attribute backlight { bool }?,
  attribute backlight_time { text }?,
  attribute backlight_cycle { text }?,
  attribute sound { text }?,
  attribute url_jump { text }?,
  attribute jump_itemid { text }?,
  attribute usemap { text }?
cell_scene = element cell_scene { cell_scene-attlist, cell_draw_image* }
cell_scene-attlist &=
  attribute position { text }?,
  attribute scroll_time { text }?,
  attribute draw_image_time { text }?,
  attribute vibration { bool }?,
  attribute display_vibration { bool }?,
  attribute vibration_direction { "vertical" | "horizontal" }?,
  attribute vibration_time { text }?,
  attribute backlight { bool }?,
  attribute backlight_time { text }?,
  attribute backlight_cycle { text }?,
  attribute sound { text }?,
  attribute url_jump { text }?
cell_draw_image =
  element cell_draw_image { cell_draw_image-attlist, empty }
cell_draw_image-attlist &=
  attribute src { text },
  attribute position { text },
  attribute speech { text }?,
  attribute speaker { text }?
page = element page { page-attlist, (page_image, speech*)+ }
page-attlist &=
  attribute bg_color { text }?,
  attribute bg_color_space { "RGB" }?,
  attribute id { text },
  attribute cell_jump_id { text }
page_image = element page_image { page_image-attlist, empty }
page_image-attlist &= attribute src { text }
speech = element speech { speech-attlist, text }
speech-attlist &= attribute speaker { text }?
local-setting-tags =
  default_ccs | page_progression_direction | window | proprietary
global-setting-tags = local-setting-tags | permission
global_setting =
  element global_setting {
    global_setting-attlist, global-setting-tags*
  }
global_setting-attlist &= empty
local_setting =
  element local_setting { local_setting-attlist, local-setting-tags* }
local_setting-attlist &= empty
default_ccs = element default_ccs { default_ccs-attlist, text }
default_ccs-attlist &= empty
page_progression_direction =
  element page_progression_direction {
    page_progression_direction-attlist, text
  }
page_progression_direction-attlist &= empty
window = element window { window-attlist, empty }
window-attlist &=
  attribute width { text }?,
  attribute height { text }?

proprietary =
  element proprietary {
    attribute vendor { text },
    (text
          | anyElement_proprietary
    )*

  }

anyElement_proprietary = element * {
```

```
                    attribute * {text}*,

                    (text

               | anyElement_proprietary

                    )*

          }*


bibliography =

  element bibliography {

    bibliography-attlist,

    (title

     | edition

     | creator

     | subject

     | description

     | publisher

     | distributor

     | contributor

     | date

     | identifier

     | source

     | language

     | relation

     | coverage

     | rights

     | price

     | comment

     | classification
     | rating
     | synopsis
     | image)*
  }
bibliography-attlist &= empty
title =
  element title {
    title-attlist, (char-or-external-char-or-ruby | comment)*
  }
title-attlist &=
  attribute reading { text }?,
  attribute type { "title" | "subtitle" | "series" | "other"}?
edition =
  element edition {
    edition-attlist, (char-or-external-char-or-ruby | comment)*
  }
edition-attlist &= attribute reading { text }?
creator =
  element creator {
    creator-attlist, (organization | person | comment)*
  }
creator-attlist &=
  attribute role {
    "designer"
    | "author"
    | "editor"
    | "translator"
    | "supervisor"
    | "photographer"
    | "illustrator"
    | "binder"
    | "planner"
    | "other"
  }?
subject = element subject { subject-attlist, (keyword | comment)* }
subject-attlist &= empty
description =
  element description {
    description-attlist, (char-or-external-char-or-ruby | comment)*
  }
description-attlist &= empty
publisher =
  element publisher {
    publisher-attlist, (organization | person | comment | code)*
  }
publisher-attlist &= attribute country { text }?
code =
  element code {
    code-attlist, (char-or-external-char-or-ruby | comment)*
  }
code-attlist &= attribute type { text }?
distributor =
  element distributor {
    distributor-attlist, (organization | person | comment | code)*
  }
distributor-attlist &= empty
contributor =
  element contributor {
    contributor-attlist, (organization | person | comment)*
  }
contributor-attlist &= attribute role { text }?
date =
  element date {
    date-attlist, (char-or-external-char-or-ruby | comment)*
  }
date-attlist &=
  attribute type { "publication" | "sale" | "revision" }?,
  [ a:defaultValue = "ISO8601" ] attribute system { text }?
identifier =
  element identifier {
    identifier-attlist, (char-or-external-char-or-ruby | comment)*
  }
identifier-attlist &= attribute type { text }
source =
  element source {
    source-attlist, (char-or-external-char-or-ruby | comment)*
  }
source-attlist &= empty
language = element language { language-attlist, text }
language-attlist &=
  attribute system { text }?
```

```
relation =

  element relation {

    relation-attlist, (char-or-external-char-or-ruby | comment)*

  }

relation-attlist &= empty

coverage =

  element coverage {

    coverage-attlist, (char-or-external-char-or-ruby | comment)*

  }

coverage-attlist &= empty

rights =

  element rights {

    rights-attlist, (char-or-external-char-or-ruby | comment)*

  }

rights-attlist &= empty

price = element price { price-attlist, (text | comment)* }

price-attlist &=

  attribute currency { text },

  attribute country { text }?

organization =

  element organization {

    organization-attlist, (name | contact | comment)*

  }

organization-attlist &= empty

person = element person { person-attlist, (name | contact | comment)* }

person-attlist &= empty

classification =
  element classification { classification-attlist, (text | comment)* }
classification-attlist &= attribute type { text }
rating = element rating { rating-attlist, (text | comment)* }
rating-attlist &= attribute type { "violent" | "adult" }?
synopsis =
  element synopsis {
    synopsis-attlist, (char-or-external-char-or-ruby | comment)*
  }
synopsis-attlist &= empty
keyword =
  element keyword {
    keyword-attlist, (char-or-external-char-or-ruby | comment)*
  }
keyword-attlist &=
  attribute type { text }?,
  attribute reading { text }?
image = element image { image-attlist, text }
image-attlist &=
  attribute type { "spine" | "front" | "thumbnail" | "other" }?
name =
  element name {
    name-attlist, (char-or-external-char-or-ruby | comment)*
  }
name-attlist &=
  attribute reading { text }?,
  attribute type { "first" | "last" | "middle" | "place" | "other" }?
contact =
  element contact {
    contact-attlist,
    (address | email | telephone | fax | website | comment)*
  }
contact-attlist &= empty
address =
  element address { address-attlist, (name | postcode | comment)* }
address-attlist &= attribute reading { text }?
comment = element comment { comment-attlist, text }
comment-attlist &= empty
email = element email { email-attlist, (text | comment)* }
email-attlist &= empty
telephone = element telephone { telephone-attlist, (text | comment)* }
telephone-attlist &= empty
fax = element fax { fax-attlist, (text | comment)* }
fax-attlist &= empty
website = element website { website-attlist, (text | comment)* }
website-attlist &= empty
postcode = element postcode { postcode-attlist, (text | comment)* }
postcode-attlist &= empty
package =
  element package {
    package-attlist, manifest, spine, special_page_link?
  }
package-attlist &= attribute version { text }?
spine = element spine { spine-attlist, itemref* }
spine-attlist &=
  attribute toc { text }?,
  attribute global_setting { text }?,
  attribute search_table { text }?,
  attribute bibliography { text }?
itemref = element itemref { itemref-attlist, empty }
itemref-attlist &= attribute idref { text }
manifest = element manifest { manifest-attlist, item* }
manifest-attlist &= attribute version { text }?
item = element item { item-attlist, permission* }
item-attlist &=
  attribute id { text },
  attribute href { text },
  attribute media-type { text }
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 type {
    "cover"
    | "title_page"
    | "preface"
    | "contents"
    | "body"
    | "column"
    | "note"
    | "figure"

    | "ad"

    | "afterword"

    | "appendix"

    | "answer"

    | "glossary"

    | "bibliography"

    | "commentary"

    | "index"

    | "imprint"

    | "author_info"

    | "other"

    | "flow_title"

  }?,

  attribute title { text }?

search_table =

  element search_table { search_table-attlist, search_table_def+ }

search_table-attlist &=

  [ a:defaultValue = "false" ] attribute bookmark { bool }?,

  [ a:defaultValue = "false" ] attribute wordbook { bool }?,

  [ a:defaultValue = "true" ] attribute jump_search_root { bool }?,

  [ a:defaultValue = "true" ] attribute jump_search { bool }?,

  [ a:defaultValue = "true" ] attribute all_search { bool }?

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 = "false" ] attribute use_default { bool }?,
  [ a:defaultValue = "implicit" ]
  attribute sorting_rule { "implicit" | "unicode" | "other" }?,
  attribute name { text }?,
  attribute short_name { text }?,
  [ a:defaultValue = "false" ] attribute wild { bool }?,
  [ a:defaultValue = "false" ] attribute blank { bool }?,
  [ a:defaultValue = "false" ] attribute end { bool }?,
  attribute help_page_id { text }?
enable_key_type =
  element enable_key_type { enable_key_type-attlist, char_list? }
enable_key_type-attlist &=
  [ a:defaultValue = "false" ] attribute numerals { bool }?,
  [ a:defaultValue = "false" ] attribute basic_alphabet { bool }?,
  [ a:defaultValue = "false" ] attribute kana { bool }?,
  [ a:defaultValue = "false" ] attribute kanji { bool }?,
  [ a:defaultValue = "false" ] attribute listed { bool }?
key_normalization =
  element key_normalization { key_normalization-attlist, empty }
key_normalization-attlist &=
  [ a:defaultValue = "true" ] attribute capitalization { bool }?,
  [ a:defaultValue = "delete" ]
  attribute cho_on { "delete" | "repeat" | "as_is" }?,
  [ a:defaultValue = "true" ] attribute daku_on { bool }?,
  [ a:defaultValue = "true" ] attribute handaku_on { bool }?,
  [ a:defaultValue = "true" ] attribute soku_on { bool }?,

  [ a:defaultValue = "true" ] attribute yo_on { bool }?,

  [ a:defaultValue = "true" ] attribute other_small_kana { bool }?

char_list = element char_list { char_list-attlist, text }

char_list-attlist &= empty

flip_animation =

  element flip_animation {

    flip_animation-attlist, audio?, flip_animation_source+

  }

flip_animation-attlist &= attribute renewal_time { text }?

flip_animation_source =

  element flip_animation_source { flip_animation_source-attlist, empty }

flip_animation_source-attlist &=

  attribute src { text },

  attribute renewal_time { text }?

start = package | bibliography | global_setting | html | flip_animation
```

Bibliography

The following documents have served as references in the preparation of this International Standard:

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

IEC 62605:2011, *Multimedia systems and equipment – Multimedia e-publishing and e-books – Interchange format for e-dictionaries*

ISO/IEC 10179:1996, *Information technology – Processing languages – Document Style Semantics and Specification Language (DSSSL)*

ISO 639-2, *Codes for the representation of names of languages – Part 2: Alpha-3 code*

ISO 639-3, *Codes for the representation of names of languages – Part 3: Alpha-3 code for comprehensive coverage of languages*

ISO 646, *Information technology – ISO 7-bit coded character set for information interchange*

ISO 3166-2, *Codes for the representation of names of countries and their subdivisions — Part 2: Country subdivision code*

ISO 4217, *Codes for the representation of currencies and funds*

ISO 8601, *Data elements and interchange formats – Information interchange – Representation of dates and times*

ISO/IEC 8859-1, *Information technology – 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1*

ISO/IEC 8859-9, *Information technology – 8-bit single-byte coded graphic character sets — Part 9: Latin alphabet No. 5*

ISO/IEC 8859-10, *Information technology – 8-bit single-byte coded graphic character sets — Part 10: Latin alphabet No. 6*

ISO/IEC 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)*

ISO 15836:2009, *Information and documentation – The Dublin Core metadata element set*

W3C Recommendation, *Extensible Markup Language (XML)* 1.0 (Third Edition), 2004-02, http://www.w3.org/TR/2004/REC-xml-20040204

W3C Recommendation, *Extensible Stylesheet Language (XSL)* Version 1.0, 2001-10, http://www.w3.org/TR/2001/REC-xsl-20011015

Internet Engineering Task Force(IETF),*"RFC 3986-Uniform Resource Identifier (URI): Generic Syntax"*, http://www.ietf.org/rfc/rfc3986.txt

XHTML1.1 – Module-based XHTML – Second Edition http://www.w3.org/TR/xhtml11/

A vocabulary and associated APIs for HTML and XHTML W3C Working Draft 25 May 2011 http://www.w3.org/TR/2011/WD-html5-20110525

CSS Text Level 3 W3C Working Draft 5 October 2010 http://www.w3.org/TR/2010/WD-css3-text-20101005/

CSS Writing Modes Module Level 3 Editor's Draft 02 December 2010 http://www.w3.org/TR/2010/WD-css3-writing-modes-20101202/

JIS X 0201:1997, 7-bit and 8-bit Coded Character Sets for Information Interchange

JIS X 0208:1997, 7-bit And 8-bit Double Byte Coded KANJI Sets for Information Interchange

JIS X 0213:2004, 7-bit And 8-bit Double Byte Coded KANJI Sets for Information Interchange

___

This page deliberately left blank

This page deliberately left blank

British Standards Institution (BSI)

BSI is the national body responsible for preparing British Standards and other standards-related publications, information and services.

BSI is incorporated by Royal Charter. British Standards and other standardization products are published by BSI Standards Limited.

About us

We bring together business, industry, government, consumers, innovators and others to shape their combined experience and expertise into standards -based solutions.

The knowledge embodied in our standards has been carefully assembled in a dependable format and refined through our open consultation process. Organizations of all sizes and across all sectors choose standards to help them achieve their goals.

Information on standards

We can provide you with the knowledge that your organization needs to succeed. Find out more about British Standards by visiting our website at bsigroup.com/standards or contacting our Customer Services team or Knowledge Centre.

Buying standards

You can buy and download PDF versions of BSI publications, including British and adopted European and international standards, through our website at bsigroup.com/shop, where hard copies can also be purchased.

If you need international and foreign standards from other Standards Development Organizations, hard copies can be ordered from our Customer Services team.

Subscriptions

Our range of subscription services are designed to make using standards easier for you. For further information on our subscription products go to bsigroup.com/subscriptions.

With **British Standards Online (BSOL)** you'll have instant access to over 55,000 British and adopted European and international standards from your desktop. It's available 24/7 and is refreshed daily so you'll always be up to date.

You can keep in touch with standards developments and receive substantial discounts on the purchase price of standards, both in single copy and subscription format, by becoming a **BSI Subscribing Member**.

PLUS is an updating service exclusive to BSI Subscribing Members. You will automatically receive the latest hard copy of your standards when they're revised or replaced.

To find out more about becoming a BSI Subscribing Member and the benefits of membership, please visit bsigroup.com/shop.

With a **Multi-User Network Licence (MUNL)** you are able to host standards publications on your intranet. Licences can cover as few or as many users as you wish. With updates supplied as soon as they're available, you can be sure your documentation is current. For further information, email bsmusales@bsigroup.com.

Revisions

Our British Standards and other publications are updated by amendment or revision.

We continually improve the quality of our products and services to benefit your business. If you find an inaccuracy or ambiguity within a British Standard or other BSI publication please inform the Knowledge Centre.

Copyright

All the data, software and documentation set out in all British Standards and other BSI publications are the property of and copyrighted by BSI, or some person or entity that owns copyright in the information used (such as the international standardization bodies) and has formally licensed such information to BSI for commercial publication and use. Except as permitted under the Copyright, Designs and Patents Act 1988 no extract may be reproduced, stored in a retrieval system or transmitted in any form or by any means – electronic, photocopying, recording or otherwise – without prior written permission from BSI. Details and advice can be obtained from the Copyright & Licensing Department.

Useful Contacts:

Customer Services
Tel: +44 845 086 9001
Email (orders): orders@bsigroup.com
Email (enquiries): cservices@bsigroup.com

Subscriptions
Tel: +44 845 086 9001
Email: subscriptions@bsigroup.com

Knowledge Centre
Tel: +44 20 8996 7004
Email: knowledgecentre@bsigroup.com

Copyright & Licensing
Tel: +44 20 8996 7070
Email: copyright@bsigroup.com

BSI Group Headquarters

389 Chiswick High Road London W4 4AL UK

bsi.

...making excellence a habit.™