

# INTERNATIONAL STANDARD

# IEC 62448

First edition  
2007-04

---

---

**Multimedia systems and equipment –  
Multimedia E-Publishing and E-Books –  
Generic format for E-Publishing**

IECNORM.COM: Click to view the full PDF of IEC 62448:2007

Withdrawing



Reference number  
IEC 62448:2007(E)



## THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2007 IEC, Geneva, Switzerland

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

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

IEC Central Office  
3, rue de Varembe  
CH-1211 Geneva 20  
Switzerland  
Email: [inmail@iec.ch](mailto:inmail@iec.ch)  
Web: [www.iec.ch](http://www.iec.ch)

### About the IEC

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

### About IEC publications

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

- Catalogue of IEC publications: [www.iec.ch/searchpub](http://www.iec.ch/searchpub)

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 It also gives information on projects, withdrawn and replaced publications.

- IEC Just Published: [www.iec.ch/online\\_news/justpub](http://www.iec.ch/online_news/justpub)

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

- Customer Service Centre: [www.iec.ch/webstore/custserv](http://www.iec.ch/webstore/custserv)

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

Email: [csc@iec.ch](mailto:csc@iec.ch)  
Tel.: +41 22 919 02 11  
Fax: +41 22 919 03 00

# INTERNATIONAL STANDARD

# IEC 62448

First edition  
2007-04

---

---

## Multimedia systems and equipment – Multimedia E-Publishing and E-Books – Generic format for E-Publishing

IECNORM.COM: Click to view the full PDF of IEC 62448:2007



Commission Electrotechnique Internationale  
International Electrotechnical Commission  
Международная Электротехническая Комиссия

PRICE CODE **XC**

*For price, see current catalogue*

## CONTENTS

FOREWORD.....	3
INTRODUCTION.....	5
1 Scope.....	6
2 Normative references.....	6
3 Terms and definitions.....	6
4 Position and requirements for generic format.....	6
4.1 Generic format in contents creation/distribution model.....	6
4.2 Requirements for generic format.....	7
5 Notation.....	7
6 Logical structure.....	7
7 Semantics.....	9
Annex A (normative) BBeB Xylog Format.....	10
Bibliography.....	80
Figure 1 – Contents creation/distribution model.....	7
Figure A.1 – Conceptual diagram of the “Block layout”.....	24
Figure A.2 – Coordinate system.....	25
Figure A.3 – Page layout composition.....	26
Figure A.3 – Page layout composition.....	26
Figure A.4 – Block layout composition.....	27
Figure A.5 – Bgimagemode attribute.....	33
Figure A.6 – Layout attribute.....	34

## INTERNATIONAL ELECTROTECHNICAL COMMISSION

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

## FOREWORD

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

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

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

Sony Corporation  
1-7-1 Konan Minato-ku  
TOKYO  
108-0075 Japan

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

International Standard IEC 62448 has been prepared by technical area 10: Multimedia e-publishing and e-books, of IEC technical committee 100: Audio, video and multimedia systems and equipment.

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

CDV	Report on voting
100/1090/CDV	100/1163/RVC

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

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

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

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

A bilingual version of this publication may be issued at a later date.

IECNORM.COM: Click to view the full PDF of IEC 62448:2007  
Withdrawn

## INTRODUCTION

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

IECNORM.COM: Click to view the full PDF of IEC 62448:2007  
Withdrawn

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

## 1 Scope

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

## 2 Normative references

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

IEC 62229:2005, *Multimedia systems and equipment – Multimedia e-publishing and e-books – Conceptual model for multimedia e-publishing*

ISO/IEC 19757-2:2003 *Information technology – Document Schema Definition Language (DSDL) – Part 2: Regular-grammar based validation – RELAX NG Amendment 1 (2003)*

ISO 639, *Codes for the representation of names of languages*

## 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

### 3.2

#### **data preparer**

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

### 3.3

#### **publisher**

organization or person that issues and distributes an e-book

### 3.4

#### **generic format**

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

## 4 Position and requirements for generic format

### 4.1 Generic format in contents creation/distribution model

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

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

IEC 2379/06

**Figure 1 – Contents creation/distribution model**

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

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

#### 4.2 Requirements for generic format

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

- a) logical structure:  
the generic format has to have a logical structure that can easily be revised;
- b) style specification:  
the logical elements in the generic format should be rendered in accordance with a style specification for creating an appropriate reader's format;
- c) heterogeneous structures:  
multimedia e-books may include heterogeneous structures such as mathematics, chemistry and music notes according to the genre of contents. Those specific structures should be imported to the generic format;
- d) metadata:  
the generic format should support some metadata to manage the e-book contents data.

### 5 Notation

The logical structure of generic format is described by using RELAX NG schema, ISO/IEC 19757-2 and its amendment 1.

### 6 Logical structure

The logical structure of generic format: ebook-g, provides a simple and extensible format ebook-g-core and an existing and actually employed format bbebylog. The structure of bbebylog is shown in Annex A.

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

```
start = ebook-g
```

```
ebook-g = ebook-g-core
```

```
    | external "bbebylog.rnc"
```

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

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

```
body-g = element body-g { body }
```

```
body =
```

```
    title*,
```

```
    foreword*,
```

introduction\*,  
titled-clause+,  
annex\*,  
bibliography\*  
title = element title {  
    element main { text },  
    element sub { text }?  
}

foreword = element foreword { text }

introduction = element introduction { block+ }

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

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

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

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

referenced-document =  
    element referenced-document {  
        id,  
        element abbrev { text },  
        element title { text },  
        element field { text }\*,  
        element url { xsd:anyURI }  
    }

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

bibliography = element bibliography { referenced-document+ }

block = p | ol | ul | example | note | pre | float

p = element p { inline }

float = table | figure

table = external "table.rnc"

figure = external "figure.rnc"

ol =  
    element ol {  
        element li { id, block+ }+  
    }

ul =  
    element ul {  
        element li { block+ }+  
    }

example = element example { p+ }

note = element note { p+ }

```
pre = element pre { pre-content }
pre-content =
  (text
   | element var { pre-content })*
inline =
  (text
   | element code | b | i | var { inline }
   | ref
   | strong)*
ref =
  element ref {
    attribute to { xsd:IDREF }
  }
id = attribute id { xsd:ID }?
strong = element strong { text }
```

## 7 Semantics

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

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

## **Annex A** (normative)

### **BBeB Xylog Format**

#### **A.1 General**

This annex specifies the bbebxlog format mentioned in Clause 6.

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

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

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

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

#### **A.2 Structure description**

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

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

DrawChar =

- (element.Plot
- | element.CR
- | element.Fill
- | element.CharButton
- | element.Yoko
- | element.Tate
- | element.Nekase
- | element.NoBR
- | element.DrawChar
- | element.Italic
- | element.Bold
- | SimpleChar1)\*

SimpleChar1 =

- (element.Rubi
- | element.Box
- | element.EmpDots
- | element.EmpLine
- | element.Sub
- | element.Sup

```

| element.Space
| SimpleChar0)*
SimpleChar0 = (text | element.Gaiji | element.AltString)*
SimpleChar2 = (element.Plot | SimpleChar0)*
# BBeB XML Xylog Basic
element.BBeBXylog =
  element BBeBXylog {
    attlist.version,
    element.BookInformation,
    element.Main,
    element.Style,
    element.Objects
  }
attlist.version &=
  attribute version { text }
  >> a:documentation [
    ' Constraint: Current "version" attribute value is "1.0" '
  ]
element.BookInformation =
  element BookInformation { element.Info, element.TOC }
element.Info =
  element Info {
    attlist.version, element.BookInfo, element.DocInfo, element.Keyword*
  }
element.BookInfo =
  element BookInfo {
    element.Title,
    element.Author,
    element.BookID,
    element.Publisher,
    element.Label,
    element.Category*,
    element.Classification,
    element.FreeText
  }
element.Title = element Title { attlist.reading, text }
attlist.reading &= attribute reading { text }
element.Author = element Author { attlist.reading, text }
element.BookID = element BookID { text }
element.Publisher = element Publisher { attlist.reading, text }
element.Label = element Label { attlist.reading, text }
element.Category = element Category { text }
element.Classification = element Classification { text }
element.FreeText = element FreeText { text }
element.DocInfo =

```

```
element DocInfo {
  element.CThumbnail,
  element.Language,
  element.Creator,
  element.SumPage
}
element.CThumbnail = element CThumbnail { attlist.CThumbnail }
attlist.CThumbnail &= attribute file { text }
element.Language =
  element Language {
    xsd:string { length = "2" }
    >> a:documentation [
      " Constraint: Set two-letter primary language code which is specified in ISO 639 "
    ]
  }
element.Creator = element Creator { text }
element.SumPage = element SumPage { text }
element.Keyword = element Keyword { text }
element.TOC = element TOC { element.TocLabel* }
element.TocLabel = element TocLabel { attlist.jump, text }
element.Main = element Main { element.Page+ }
element.Page =
  element Page {
    attlist.Page,
    element.Common.Page
  }
  >> sch:pattern [
    name = "Style Check"
    "\x{a}" ~
    " "
    sch:rule [
      context = "Page"
      "\x{a}" ~
      " "
      sch:assert [
        test = "@pagestyle = //Style/PageStyle/@stylelabel"
        "No PageStyle"
      ]
    ]
    "\x{a}" ~
    " "
  ]
  "\x{a}" ~
  " "
}
element.Common.Page =
```

```

(element.Common.Objects
 | element.BlockSpace
 | element.RuledLine
 | element.Wait)*
attlist.id &= attribute objid { xsd:string }
attlist.refid &= attribute refobj { xsd:string }
attlist.refstream &= attribute refstream { xsd:string }
attlist.Page &=
  attlist.id,
  attribute pagestyle { text },
  attlist.Common.Page
element.TextBlock =
element TextBlock {
  attlist.TextBlock,
  ((element.P | element.CR)*)
  >> sch:pattern [
    name = "Style Check"
    "\x{a}" ~
    " "
    sch:rule [
      context = "TextBlock"
      "\x{a}" ~
      " "
      sch:assert [
        test = "@blockstyle = //Style/BlockStyle/@stylelabel"
        "No BlockStyle"
      ]
      "\x{a}" ~
      " "
    ]
    "\x{a}" ~
    " "
  ]
  >> sch:pattern [
    name = "Style Check"
    "\x{a}" ~
    " "
    sch:rule [
      context = "TextBlock"
      "\x{a}" ~
      " "
      sch:assert [
        test = "@textstyle = //Style/TextStyle/@stylelabel"
        "No TextStyle"
      ]
    ]
  ]

```

```

        "\x{a}" ~
        " "
    ]
    "\x{a}" ~
    " "
]
}
attlist.TextBlock &=
    attlist.id,
    attribute textstyle { text },
    attribute blockstyle { text },
    attlist.Common.Text,
    attlist.Common.Block
element.P = element P { attlist.P, DrawChar }
attlist.P &= attribute refesound { xsd:string }?
element.Plot = element Plot { attlist.Plot, text }
attlist.xsize &= attribute xsize { xsd:unsignedShort }
attlist.ysize &= attribute ysize { xsd:unsignedShort }
attlist.Plot &=
    attlist.xsize,
    attlist.ysize,
    attlist.refid,
    [ a:defaultValue = "bottom" ]
    attribute adjustment { "center" | "baseline" | "top" | "bottom" }?
element.CR = element CR { empty }
element.Fill =
    element Fill {
        attribute code { text },
        empty
    }
element.Space = element Space { attlist.xsize, empty }
element.CharButton = element CharButton { attlist.refid, SimpleChar1 }
element.Rubi = element Rubi { (element.Oyamoji, element.Rubimoji)+ }
element.Oyamoji = element Oyamoji { SimpleChar0 }
element.Gaiji = element Gaiji { attlist.Gaiji, text }
attlist.Gaiji &=
    attlist.refid,
    attribute fontfacename { text }?,
    attribute code { text }
element.AltString = element AltString { element.Org, element.Alt }
element.Org = element Org { text }
element.Alt = element Alt { text }
element.Rubimoji = element Rubimoji { SimpleChar0 }
element.Box = element Box { attlist.Box, SimpleChar0 }
attlist.linetype &=

```

```

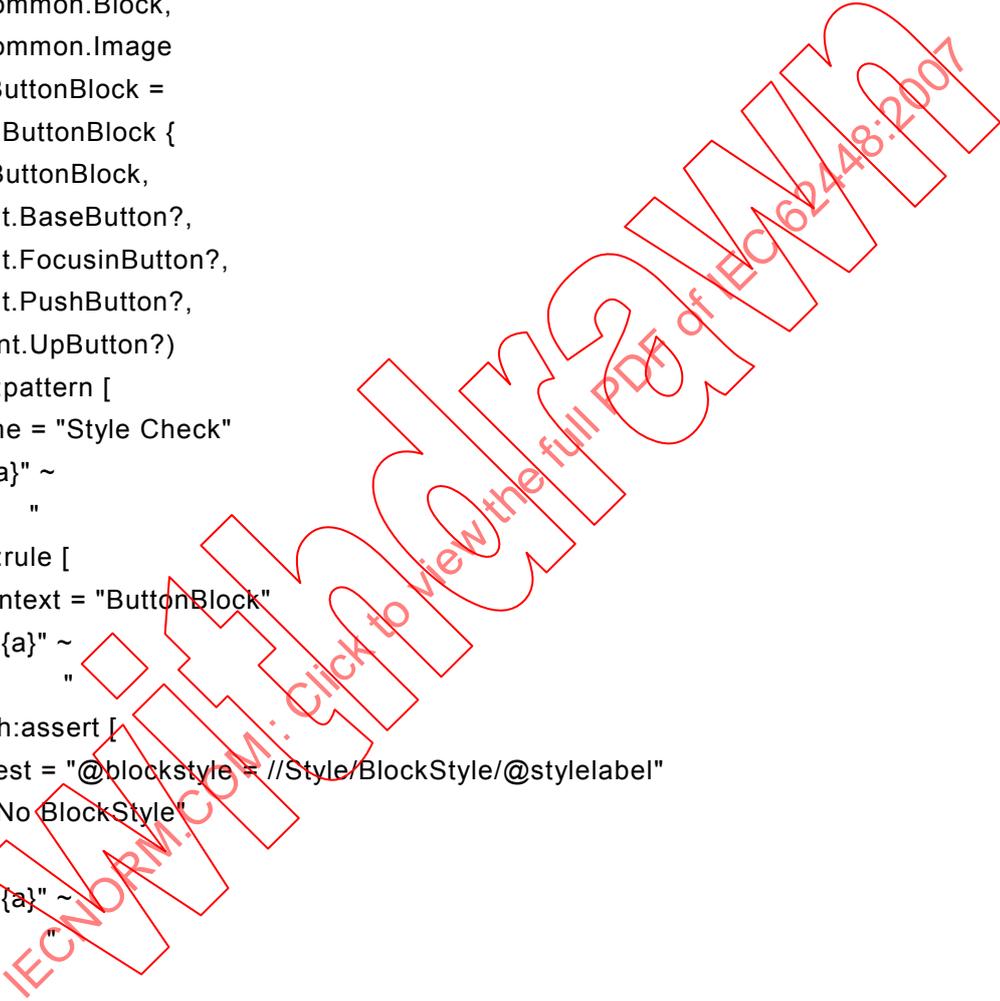
    attribute linetype { "solid" | "dotted" | "dashed" | "double" }
attlist.Box &= attlist.linetype?
element.EmpDots = element EmpDots { attlist.EmpDots, SimpleChar0 }
attlist.Empdotsposition &=
    attribute empdotsposition { "before" | "after" }
attlist.EmpdotsChar &=
    attribute empdotscode { text },
    attribute empdotsfontname { text },
    attribute refempdotsfont { text }
attlist.EmpDots &= attlist.Empdotsposition?, attlist.EmpdotsChar?
element.EmpLine = element EmpLine { attlist.EmpLine, SimpleChar0 }
attlist.Emplineposition &=
    attribute emplineposition { "before" | "after" }
attlist.Emplintype &=
    attribute emplintype {
        "none" | "solid" | "dotted" | "dashed" | "double"
    }
attlist.EmpLine &= attlist.Emplintype?, 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 Block Style"
      ]
    ]
    "\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" | "TbRI" }  
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" | "RI" },  
    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" | "file" | "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.

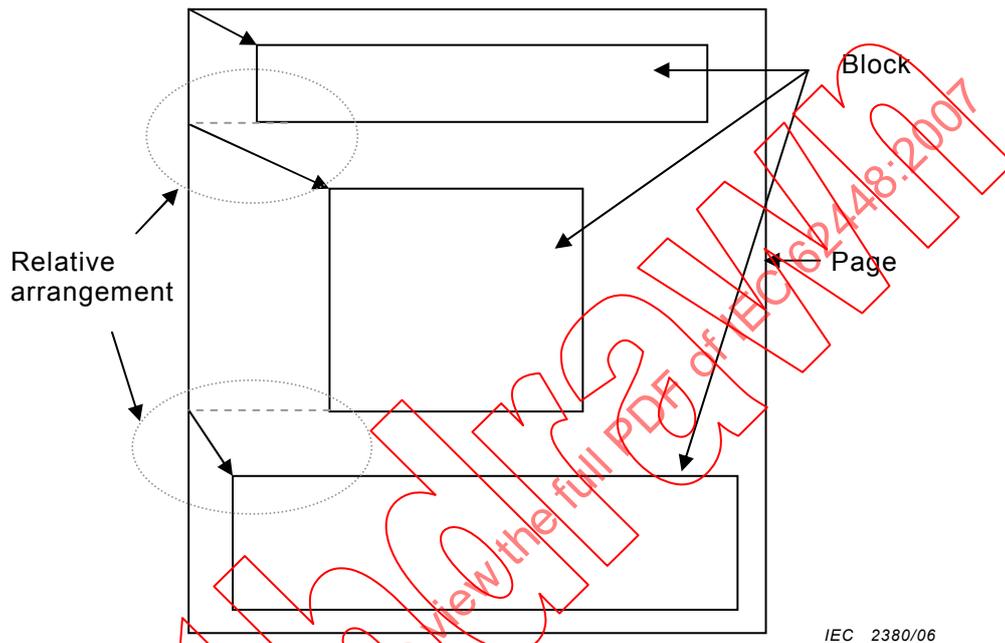


Figure A.1 – Conceptual diagram of the “Block layout”

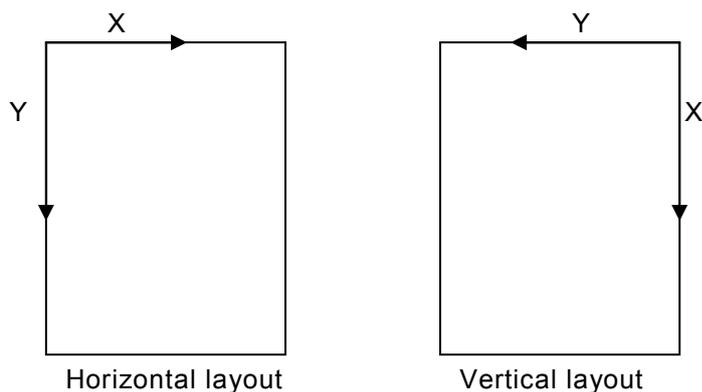
#### A.3.2 Block

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

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

#### A.3.3 Coordinate system

This specification uses a two-dimensional coordinate system (X,Y) that depends on the vertical layout or the horizontal layout.



IEC 2381/06

Figure A.2 – Coordinate system

## A.4 Terminology

### A.4.1 Page layout

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

Screen size:

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

topmargin:

the space between the top edge of screen and Header area

Header area:

area where header(s) is/are placed

headheight:

the height of Header area

headsep:

the space between Header area and Main text area

Main text area:

area where main content is placed

textheight:

the height of Main text area

textwidth:

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

footspace:

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

footheight:

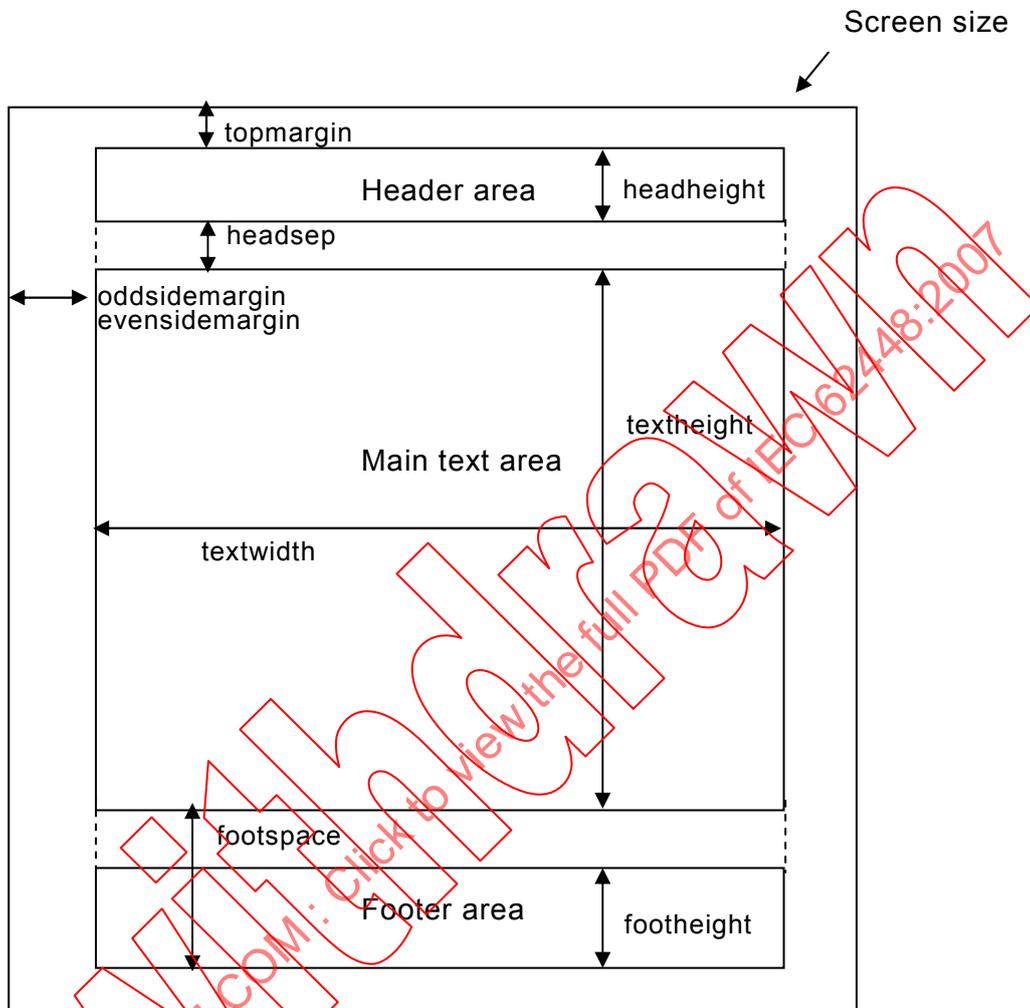
the height of Footer area

oddsidemargin:

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

evensidemargin:

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



IEC 2382/06

**Figure A.3 – Page layout composition**

**A.4.2 Block layout**

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

rendering area:

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

topskip:

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

sidemargin:

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

footskip:

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

blockheight:

height of the “Block”

blockwidth:

width of the "Block"

parindent:

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

parskip:

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

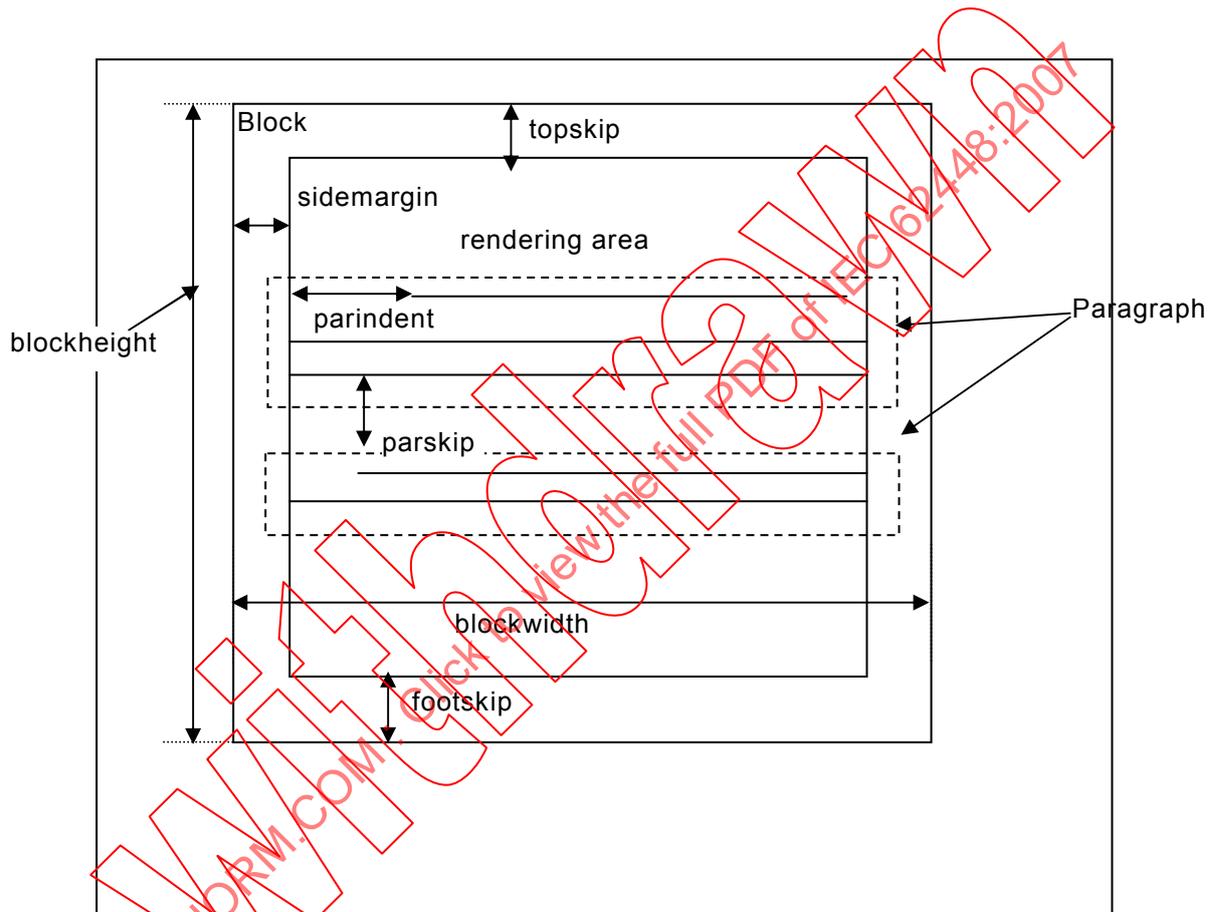


Figure A.4 – Block layout composition

IEC 2383/06

### A.4.3 Data types

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

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

## A.5 Elements and attributes

### A.5.1 BBeBXylog

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

Attribute name	Default value	Data type	Comments
<b>version</b> (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.

Attribute name	Default value	Data type	Comments
<b>version</b> (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.

Attribute name	Default value	Data type	Comments
<b>reading</b> (required)		String for sorting	

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

#### A.5.6 Author

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

Attribute name	Default value	Data type	Comments
<b>reading</b> (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.

Attribute name	Default value	Data type	Comments
<b>reading</b> (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.

Attribute name	Default value	Data type	Comments
<b>reading</b> (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.

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

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

#### A.5.15 Language

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

#### A.5.16 Creator

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

### A.5.17 SumPage

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

### A.5.18 Keyword

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

### A.5.19 TOC

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

### A.5.20 TocLabel

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

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

#### A.5.20.1 refobj

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

#### A.5.20.2 refpage

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

### A.5.21 Main

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

### A.5.22 Page

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

Attribute name	Default value	Data type	Comments
<b>bgimagemode</b>		String	Specifies “fix”, “tile” or “centering”
<b>evenfooterid</b>		Object ID string	Specifies the “objid” of the <b>Footer</b> element
<b>evenheaderid</b>		Object ID string	Specifies the “objid” of the <b>Header</b> element
<b>evensidemargin</b>	(Unsigned integer)	Decimal string	[dot]
<b>footheight</b>	(Signed integer)	Decimal string	[dot]
<b>footspace</b>	(Signed integer)	Decimal string	[dot]
<b>headheight</b>	(Signed integer)	Decimal string	[dot]
<b>headsep</b>	(Signed integer)	Decimal string	[dot]

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

#### A.5.22.1 pagestyle

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

#### A.5.22.2 refbgimage

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

#### A.5.22.3 pageposition

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

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

The **bindingdirection** attribute “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.

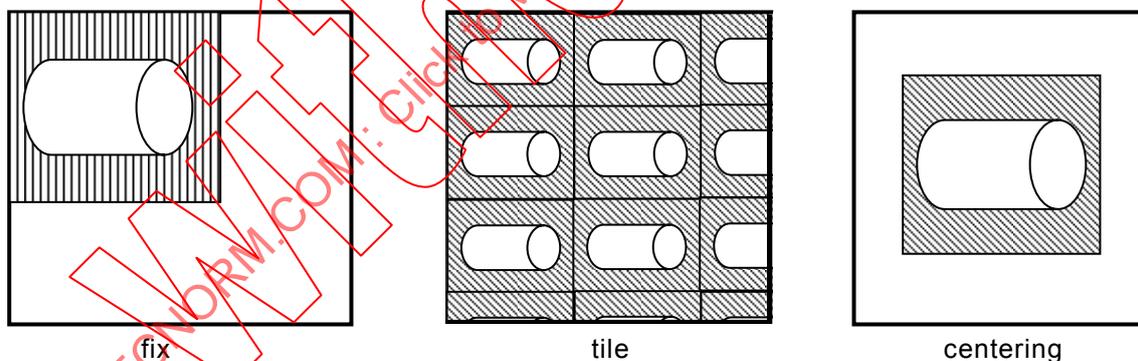
**A.5.22.8 bgimagemode**

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

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

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

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



IEC 2384/06

**Figure A.5 – Bgimagemode attribute**

**A.5.22.9 evensidemargin**

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

**A.5.22.10 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**

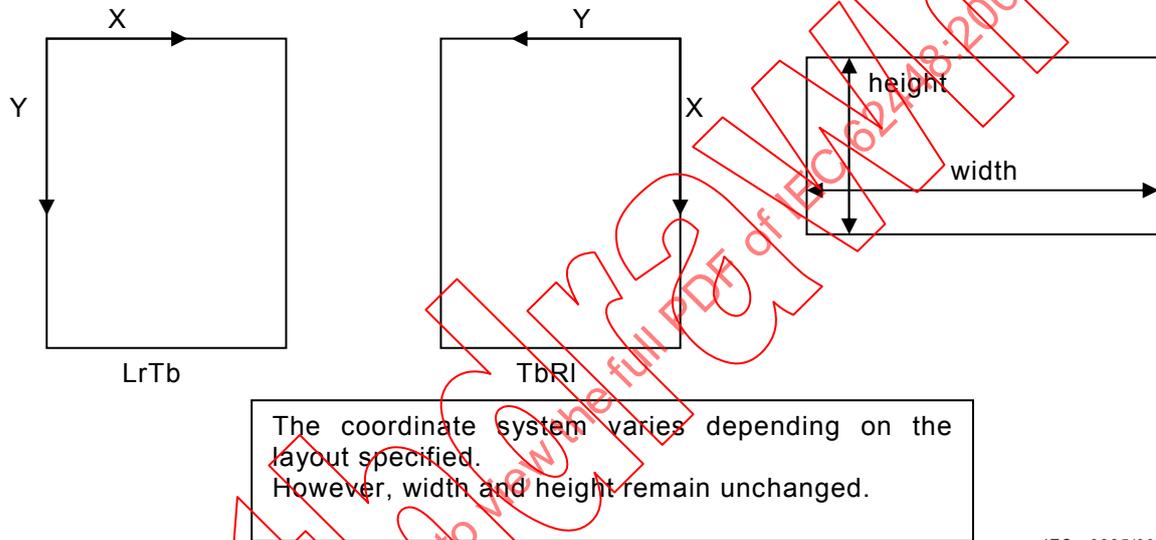
The **headheight** attribute sets the “headheight” of the “Page layout composition”.

**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.



IEC 2385/06

**Figure A.6 – Layout attribute**

**A.5.22.15 oddsidemargin**

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

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

**A.5.22.16 textheight**

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

**A.5.22.17 textwidth**

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

**A.5.22.18 topmargin**

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

**A.5.22.19 objid**

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

### A.5.23 Wait

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

Attribute name	Default value	Data type	Comments
<b>time</b> (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 ( $\times 100$  ms). When the time is a negative value, it keeps waiting for the input during the absolute value of the set time( $\times 100$  ms).

### A.5.24 TextBlock

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

Attribute name	Default value	Data type	Comments
<b>align</b>		String	Specifies "head", "center" or "foot"
<b>baselineskip</b>	(Signed integer)	Decimal string	[pt]*10
<b>bgcolor</b>		COLORREF string	
<b>bgimagemode</b>		String	Specifies "fix", "tile" or "centering". Moreover, it is necessary to specify the <b>refbgimage</b> attribute at the same time
<b>blockheight</b>	(Unsigned integer)	Decimal string	[dot]
<b>blockrule</b>		String	Specifies "horz-fixed", "horz-adjustable", "vert-fixed" or "vert-adjustable" as the <b>blockrule</b> attribute of the <b>TextBlock</b> element
<b>blockstyle</b> (Required)		Style string	It is necessary to match the specified style character string to "stylelabel" of the referred <b>BlockStyle</b> element
<b>blockwidth</b>	(Unsigned integer)	Decimal string	[dot]
<b>charspace</b>	(Signed integer)	Decimal string	[pt]*10
<b>column</b>		Decimal string	Specifies an integer between "1" and "9"
<b>columnsep</b>	(Signed integer)	Decimal string	[pt]*10
<b>empdotscode</b>		Hexadecimal string	Specifies the character code used as an emphasis dot. Moreover, it is necessary to specify the <b>empdotsfontname</b> and <b>refempdotsfont</b> attribute at the same time
<b>empdotsposition</b>		String	Specifies "before" or "after"
<b>emlinetype</b>		String	Specifies "none", "solid", "dotted", "double" or "dashed"
<b>emlineposition</b>		String	Specifies "before" or "after"
<b>fontescapement</b>		Decimal string	Specifies "0" or "2700"
<b>fontfacename</b>		String	Specifies the font name
<b>fontorientation</b>		Decimal string	Specifies "0" or "2700"
<b>fontsize</b>	(Signed integer)	Decimal string	[pt] * 10

Attribute name	Default value	Data type	Comments
<b>fontweight</b>		Decimal string	Specifies a value between "1" and "1000"
<b>fontwidth</b>	(Signed integer)	Decimal string	[pt]*10 Specifies "-10" when not changing the font shape
<b>footskip</b>	(Unsigned integer)	Decimal string	[dot]
<b>framecolor</b>		COLORREF string	
<b>framemode</b>		String	Specifies "curve" or "square"
<b>framewidth</b>	(Unsigned integer)	Decimal string	[dot]
<b>layout</b>		String	Specifies "LrTb" or "TbRl"
<b>letterspace</b>	(Signed integer)	Decimal string	[pt]*10
<b>linecolor</b>		COLORREF string	
<b>linespace</b>	(Signed integer)	Decimal string	[pt]*10
<b>textlinewidth</b>	(Unsigned integer)	Decimal string	[pt]*10
<b>objid</b> (Required)		String	Specifies the only character string in the file
<b>parindent</b>	(Signed integer)	Decimal string	[pt]*10
<b>parskip</b>	(Signed integer)	Decimal string	[pt]*10
<b>refbgimage</b>		Object ID string	Specifies the "objid" of the <b>Image</b> element. Moreover, it is necessary to specify the <b>refimagemode</b> attribute at the same time
<b>rubyalign</b>		String	Specifies "start" or "center"
<b>rubyadjust</b>		String	Specifies the "line-edge" or "none" character string
<b>rubyoverhang</b>		String	Specifies the "auto" or "none" character string
<b>sidemargin</b>	(Unsigned integer)	Decimal string	[dot]
<b>textbgcolor</b>		COLORREF string	
<b>textcolor</b>		COLORREF string	
<b>textstyle</b> (Required)		Style string	It is necessary to match the specified style character string to "stylelabel" of the referred <b>TextStyle</b> element
<b>toclabel</b>			Specifies the character string used in the TOC list
<b>topskip</b>	(Unsigned integer)	Decimal string	[dot]
<b>wordspace</b>	(Signed integer)	Decimal string	[pt]*10
<b>empdotsfontname</b>		String	Specifies the font name of the font used for emphasis dots. Moreover, it is necessary to specify the <b>empdotscode</b> and <b>refempdotsfont</b> attribute at the same time
<b>refempdotsfont</b>		Object ID String	Specifies the "objid" of the Font element. Moreover, it is necessary to specify the <b>empdotscode</b> and <b>empdotsname</b> 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 “2700” 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 “2700” 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.

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”.

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.

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.

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 emlineposition**

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

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

**A.5.24.21 emlinetype**

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

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.

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”.

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

**A.5.24.34 blockstyle**

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

**A.5.24.35 blockwidth**

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

**A.5.24.36 footskip**

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

**A.5.24.37 framecolor**

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

**A.5.24.38 framemode**

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

**A.5.24.39 framewidth**

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

**A.5.24.40 layout**

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

**A.5.24.41 refbgimage**

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

**A.5.24.42 sidemargin**

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

**A.5.24.43 toclabel**

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

**A.5.24.44 topskip**

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

**A.5.24.45 empdotsfontname**

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

**A.5.24.46 refempdotsfont**

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

**A.5.25 P**

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

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

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

### A.5.26 Plot

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

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

Attribute name	Default value	Data type	Comments
<b>code</b> (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.

Attribute name	Default value	Data type	Comments
<b>xsize</b> (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.

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

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

#### A.5.31 Rubi

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

#### A.5.32 Oyamoji

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

#### A.5.33 Gaiji

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

Attribute Name	Default Value	Data Type	Comments
<b>code</b> (required)		Hexadecimal string	Can be specified between “0x0020” and “0xffff”
<b>fontfacename</b> (required)		String	Specifies the font name
<b>refobj</b> (required)		Object ID string	Specifies the “objid” of the <b>Font</b> 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 customer 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.

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

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

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

### A.5.39 EmpDots

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

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

#### A.5.39.1 empdotsposition

Refer to A.5.24.18.

#### A.5.39.2 empdotscode

Refer to A.5.24.19.

#### A.5.39.3 empdotsfontname

Refer to A.5.24.45.

#### A.5.39.4 refempdotsfont

Refer to A.5.24.46.

### A.5.40 EmpLine

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

Attribute name	Default value	Data type	Comments
<b>emplinetype</b>		String	Specifies "none", "solid", "dotted", "dashed" or "double"
<b>emplineposition</b>		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 **Sub** 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.

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

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

**A.5.48 Italic**

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

**A.5.49 Bold**

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

**A.5.50 ImageBlock**

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

Attribute name	Default value	Data type	Comments
<b>x0</b> (required)	(Unsigned integer)	Decimal string	[dot]
<b>y0</b> (required)	(Unsigned integer)	Decimal string	[dot]
<b>x1</b> (required)	(Unsigned integer)	Decimal string	[dot]
<b>y1</b> (required)	(Unsigned integer)	Decimal string	[dot]
<b>xsize</b> (required)	(Unsigned integer)	Decimal string	[dot]
<b>ysize</b> (required)	(Unsigned integer)	Decimal string	[dot]
<b>refstream</b>		Object ID string	Specifies the "objid" of the <b>ImageStream</b> element. Only "refstream" can be selected in the <b>Image</b> element
<b>objid</b> (required)		String	Specifies the only character string in the file
<b>bgcolor</b>		COLORREF string	
<b>bgimagemode</b>		String	Specifies "fix", "file" or "centering". Moreover, it is necessary to specify the <b>refbimage</b> attribute at the same time
<b>blockheight</b>	(Unsigned integer)	Decimal string	[dot]
<b>blockrule</b>		String	"block-fixed" or "block-adjustable" is specified for the ImageBlock blockrule attribute
<b>blockstyle</b> (required)		Style string	It is necessary to match the specified style character string to "stylelabel" of the referred <b>BlockStyle</b> element
<b>blockwidth</b>	(Unsigned integer)	Decimal string	[dot]
<b>footskip</b>	(Unsigned integer)	Decimal string	[dot]
<b>framecolor</b>		COLORREF string	
<b>framemode</b>		String	Specifies "curve" or "square"
<b>framewidth</b>	(Unsigned integer)	Decimal string	[dot]
<b>layout</b>		String	Specifies "LrTb" or "TbRI"
<b>refbimage</b>		Object ID string	Specifies the "objid" of the <b>Image</b> element. However, own ID cannot be specified
<b>sidemargin</b>	(Unsigned integer)	Decimal string	[dot]
<b>topskip</b>	(Unsigned integer)	Decimal string	[dot]

#### A.5.50.1 x0

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

#### A.5.50.2 y0

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

**A.5.50.3 x1**

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

**A.5.50.4 y1**

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

**A.5.50.5 xsize**

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

**A.5.50.6 ysize**

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

**A.5.50.7 refstream**

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

**A.5.50.8 objid**

Refer to A.5.22.19.

**A.5.50.9 bgcolor**

Refer to A.5.24.30.

**A.5.50.10 bgimagemode**

Refer to A.5.24.31.

**A.5.50.11 blockheight**

Refer to A.5.24.32.

**A.5.50.12 blockrule**

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

Regarding the table of value, refer to A.5.24.33

**A.5.50.13 blockstyle**

Refer to A.5.24.34.

**A.5.50.14 blockwidth**

Refer to A.5.24.35.

**A.5.50.15 footskip**

Refer to A.5.24.36.

**A.5.50.16 framecolor**

Refer to A.5.24.37.

**A.5.50.17 framemode**

Refer to A.5.24.38.

**A.5.50.18 framewidth**

Refer to A.5.24.39.

**A.5.50.19 layout**

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

**A.5.50.20 refbgimage**

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

**A.5.50.21 sidemargin**

Refer to A.5.24.42.

**A.5.50.22 topskip**

Refer to A.5.24.44.

**A.5.51 ButtonBlock**

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

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

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

**A.5.51.1 objid**

Refer to A.5.22.19.

**A.5.51.2 bgcolor**

Refer to A.5.24.30.

**A.5.51.3 bgimagemode**

Refer to A.5.51.3.

**A.5.51.4 blockheight**

Refer to A.5.24.32.

**A.5.51.5 blockrule**

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

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

**A.5.51.6 blockstyle**

Refer to A.5.24.34.

**A.5.51.7 blockwidth**

Refer to A.5.24.35.

**A.5.51.8 footskip**

Refer to A.5.24.36.

**A.5.51.9 framecolor**

Refer to A.5.24.37.

**A.5.51.10 framemode**

Refer to A.5.24.38.

**A.5.51.11 framewidth**

Refer to A.5.24.39.

**A.5.51.12 layout**

The **layout** attribute specifies the coordinate system of the **ButtonBlock** element. When "LrTb" is specified, the origin is in the top left, the X-axis is toward the right and the Y-axis is toward the bottom. When "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 **refbgimage** attribute specifies the "objid" of the **Image** element to be shown as the background of the **ButtonBlock** element. No image is shown if the "objid" is not specified.

**A.5.51.14 sidemargin**

Refer to A.5.24.42.

**A.5.51.15 topskip**

Refer to A.5.24.44.

**A.5.52 BaseButton**

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

Attribute name	Default value	Data type	Comments
<b>refimage</b>		Object ID string	Specifies the "objid" of the <b>Image</b> 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.

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

Refer to A.5.52.

**A.5.54 JumpTo**

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

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

**A.5.54.1 refpage**

Refer to A.5.20.2.

**A.5.54.2 refobj**

Refer to A.5.20.1.

**A.5.55 Run**

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

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

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.

Attribute name	Default value	Data type	Comments
refimage		Object ID string	Specifies the “objid” of the <b>Image</b> 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.

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

Refer to A.5.52.

#### A.5.60 Canvas

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

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

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

**A.5.60.10 framemode**

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

**A.5.61 PutObj**

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

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

**A.5.61.1 x1**

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

**A.5.61.2 y1**

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

**A.5.61.3 refobj**

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

**A.5.62 Moveto**

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

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

**A.5.62.1 x1**

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

**A.5.62.2 y1**

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

**A.5.63 Lineto**

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

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

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

#### A.5.64 DrawBox

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

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

##### A.5.64.1 x1

Refer to A.5.63.1.

##### A.5.64.2 y1

Refer to A.5.63.2.

##### A.5.64.3 linewidth

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

##### A.5.64.4 linecolor

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

##### A.5.64.5 linetype

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

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

##### A.5.64.6 fillcolor

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

### A.5.65 DrawEllipse

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

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

#### A.5.65.1 x1

Refer to A.5.63.1.

#### A.5.65.2 y1

Refer to A.5.63.2.

#### A.5.65.3 linewidth

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

#### A.5.65.4 linecolor

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

#### A.5.65.5 linetype

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

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

#### A.5.65.6 fillcolor

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

### A.5.66 RuledLine

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

Attribute name	Default value	Data type	Comments
<b>linelength</b> (required)	(Unsigned integer)	Decimal string	[dot]
<b>linetype</b> (required)		string	Specifies "solid", "dotted", "dashed", or "double"
<b>linewidth</b> (required)	(Unsigned integer)	Decimal string	[dot]
<b>linecolor</b>	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

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

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

#### A.5.68.1 stylelabel

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

#### A.5.68.2 objid

Refer to A.5.22.19.

#### A.5.69 SetDefault

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

Attribute name	Default value	Data type	Comments
----------------	---------------	-----------	----------

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

Refer to A.5.24.20.

**A.5.69.7 emlinetype**

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.

Wait flag value	Operation specification
replay	Replay
noreplay	No replay

**A.5.70 RegistFont**

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

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

**A.5.70.1 fontfilename**

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

**A.5.70.2 file**

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

**A.5.70.3 fontname**

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

**A.5.70.4 encoding**

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

**A.5.71 BookSetting**

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

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

#### A.5.71.1 bindingdirection

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

#### A.5.71.2 dpi

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

#### A.5.71.3 screenwidth

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

#### A.5.71.4 screenheight

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

#### A.5.71.5 colordepth

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

#### A.5.72 TextStyle

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

Attribute name	Default value	Data type	Comments
<b>objid</b> (required)		String	Specifies the only character string in the file
<b>stylelabel</b> (required)		Style string	Specifies the style character string
<b>Fontsize</b>	"0" (Signed integer)	Decimal string	[pt]*10
<b>Fontwidth</b>	"-10" (Signed integer)	Decimal string	[pt]*10 Specifies "-10" when not changing the font shape
<b>Fontescapement</b>	"0"	Decimal string	Specifies "0" or "2700"
<b>Fontorientation</b>	"0"	Decimal string	Specifies "0" or "2700"

Attribute name	Default value	Data type	Comments
<b>fontfacename</b> (required)		String	Specifies the font name
<b>Textcolor</b>	"0x00000000"	COLORREF string	
<b>Textbgcolor</b>	"0xff000000"	COLORREF string	
<b>Wordspace</b>	"0" (Signed integer)	Decimal string	[pt]*10
<b>Letterspace</b>	"0" (Signed integer)	Decimal string	[pt]*10
<b>Baselineskip</b>	"0" (Signed integer)	Decimal string	[pt]*10
<b>Linespace</b>	"0" (Signed integer)	Decimal string	[pt]*10
<b>Parindent</b>	"0" (Signed integer)	Decimal string	[pt]*10
<b>Parskip</b>	"0" (Signed integer)	Decimal string	[pt]*10
<b>Rubyalign</b>		String	Specifies the "start" or "center" character string
<b>Rubyadjust</b>		String	Specifies the "line-edge" or "none" character string
<b>Rubyoverhang</b>		String	Specifies the "auto" or "none" character string
<b>Empdotsposition</b>		String	Specifies "before" or "after"
<b>Empdotscode</b>		String	Specifies the character code used as an emphasis dot. Moreover, it is necessary to specify the <b>empdotfontname</b> and <b>refempdotfont</b> attribute at the same time
<b>Emplinesposition</b>		String	Specifies "before" or "after"
<b>Emplinetype</b>		String	Specifies the line mode value as "none", "solid", "dotted", "dashed", or "double"
<b>Column</b>	"1"	Decimal string	Specifies an integer value from "1" to "9"
<b>Columnsep</b>	"0"	Decimal string	[pt]*10
<b>Align</b>	"head"	String	Specifies "head", "center", or "foot"
<b>Textlinewidth</b>	"0" (Unsigned integer)	Decimal string	[pt]*10
<b>Linecolor</b>	"0x00000000"	COLORREF string	
<b>Charspace</b>	"0" (Signed integer)	Decimal string	[pt]*10
<b>Fontweight</b>	"400"	Decimal string	Specifies a value from "1" to "1,000"
<b>empdotfontname</b>		String	Specifies the font name of the font used for emphasis dots. Moreover, it is necessary to specify the <b>empdotscode</b> and <b>refempdotfont</b> attribute at the same time
<b>Refempdotfont</b>		Object ID string	Specifies the "objid" of the <b>Font</b> element. Moreover, it is necessary to specify the <b>empdotscode</b> and <b>empdotfontname</b> 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.