

DIN EN 62448

ICS 35.240.30

Einsprüche bis 2013-12-04
Vorgesehen als Ersatz für
DIN EN 62448:2010-03

Entwurf

**Multimediasysteme und -geräte –
E-Publishing und E-Books für Multimedia-Anwendungen –
Allgemeines Format für E-Publishing
(IEC 100/2053/CDV:2012);
Deutsche Fassung FprEN 62448:2012**

Multimedia systems and equipment –
Multimedia e-publishing und e-books –
Generic format for e-publishing
(IEC 100/2053/CDV:2012);
German version FprEN 62448:2012

Systèmes et équipements multimedia –
Publication et livres électroniques multimedia –
Format générique pour la publication électronique
(CEI 100/2053/CDV:2012);
Version allemande FprEN 62448:2012

Anwendungswarnvermerk

Dieser Norm-Entwurf mit Erscheinungsdatum 2013-10-04 wird der Öffentlichkeit zur Prüfung und Stellungnahme vorgelegt.

Weil die beabsichtigte Norm von der vorliegenden Fassung abweichen kann, ist die Anwendung dieses Entwurfes besonders zu vereinbaren.

Stellungnahmen werden erbeten

- vorzugsweise online im Norm-Entwurfs-Portal des DIN unter www.entwuerfe.din.de bzw. für Norm-Entwürfe der DKE auch im Norm-Entwurfs-Portal der DKE unter www.entwuerfe.normenbibliothek.de, sofern dort wiedergegeben;
- oder als Datei per E-Mail an dke@din.de möglichst in Form einer Tabelle. Die Vorlage dieser Tabelle kann im Internet unter www.din.de/stellungnahme oder für Stellungnahmen zu Norm-Entwürfen der DKE unter www.dke.de/stellungnahme abgerufen werden;
- oder in Papierform an die DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE (Hausanschrift: Stresemannallee 15, 60596 Frankfurt am Main).

Die Empfänger dieses Norm-Entwurfs werden gebeten, mit ihren Kommentaren jegliche relevanten Patentrechte, die sie kennen, mitzuteilen und unterstützende Dokumentationen zur Verfügung zu stellen.

Gesamtumfang 653 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE

Nationales Vorwort

Die Deutsche Fassung des europäischen Dokuments FprEN 62448:2012 „Multimediasysteme und -geräte – E-Publishing und E-Books für Multimedia-Anwendungen – Allgemeines Format für E-Publishing“ (Entwurf in der Umfrage) ist unverändert in diesen Norm-Entwurf übernommen worden.

Die Internationale Elektrotechnische Kommission (IEC) und das Europäische Komitee für Elektrotechnische Normung (CENELEC) haben vereinbart, dass ein auf IEC-Ebene erarbeiteter Entwurf für eine Internationale Norm zeitgleich (parallel) bei IEC und CENELEC zur Umfrage (CDV-Stadium) und Abstimmung als FDIS (en: Final Draft International Standard) bzw. Schluss-Entwurf für eine Europäische Norm gestellt wird, um eine Beschleunigung und Straffung der Normungsarbeit zu erreichen. Dem entsprechend ist das internationale Dokument IEC 100/2053/CDV:2012 „Multimedia systems and equipment – Multimedia e-publishing and e-books – Generic format for e-publishing“ unverändert in den Entwurf FprEN 62448:2012 übernommen worden.

Da die Deutsche Fassung noch nicht endgültig mit der Englischen und Französischen Fassung abgeglichen ist, ist die englische Originalfassung des IEC-CDV entsprechend der diesbezüglich durch die IEC erteilten Erlaubnis beigefügt. Die Nutzungsbedingungen für den deutschen Text des Norm-Entwurfes gelten gleichermaßen auch für den englischen IEC-Text.

Das internationale Dokument wurde vom TC 100 „Audio, video and multimedia systems and equipment“ der Internationalen Elektrotechnischen Kommission (IEC) erarbeitet.

Bei der Abstimmung zu dem Europäischen Schluss-Entwurf bei CENELEC und dem Internationalen Schluss-Entwurf bei IEC [Final Draft International Standard (FDIS)] sind jeweils nur „JA/NEIN“-Entscheidungen möglich, wobei „NEIN“-Entscheidungen fundiert begründet werden müssen. Dokumente, die bei CENELEC als Europäische Norm angenommen und ratifiziert werden, sind unverändert als Deutsche Normen zu übernehmen.

Für dieses Dokument ist das nationale Arbeitsgremium K 742 „Audio-, Video- und Multimediasysteme, -geräte und -komponenten“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE (www.dke.de) zuständig.

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf eine Norm ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils neueste gültige Ausgabe der in Bezug genommenen Norm.

Für den Fall einer datierten Verweisung im normativen Text bezieht sich die Verweisung immer auf die in Bezug genommene Ausgabe der Norm.

Der Zusammenhang der zitierten Normen mit den entsprechenden Deutschen Normen ergibt sich, soweit ein Zusammenhang besteht, grundsätzlich über die Nummer der entsprechenden IEC-Publikation. Beispiel: IEC 60068 ist als EN 60068 als Europäische Norm durch CENELEC übernommen und als DIN EN 60068 ins Deutsche Normenwerk aufgenommen.

Änderungen

Gegenüber DIN EN 62448:2010-03 wurden folgende Änderungen vorgenommen:

- a) Abschnitt 6 wurde geändert;
- b) Anhang B wurde aktualisiert;
- c) Anhang C wurde neu hinzugefügt.

**Multimediasysteme und -geräte –
E-Publishing und E-Books für Multimedia-Anwendungen –
Allgemeines Format für E-Publishing**

Inhalt	Seite
Einleitung	5
1 Anwendungsbereich	6
2 Normative Verweisungen	6
3 Begriffe	6
4 Stellung und Anforderungen für das allgemeine Format	7
4.1 Allgemeines Format des Modells für das Erstellen und Verteilen von Inhalten	7
4.2 Anforderungen für das allgemeine Format	7
5 Notation	7
6 Logische Struktur	7
7 Semantiken	9
Anhang A (normativ) BBeB Xylog-Format	10
A.1 Allgemeines	10
A.2 Beschreibung der Struktur	10
A.3 Block layout	19
A.4 Terminologie	20
A.5 Elemente und Attribute	23
A.6 Reservierte Elemente	75
Anhang B (normativ) XMDF-XML-Format	78
B.1 Allgemeines	78
B.2 Überblick über die Formatstruktur	78
B.3 Elemente und Attribute	79
B.4 Einzelheiten des Beschreibungsformats	90
B.5 Farbnamen (normativ)	189
B.6 Ortsbezogener Teil (informativ)	191
B.7 Festlegung des XMDF-XML-Formats in der RELAX-NG-Kompaktsyntax	192
Anhang C (normativ) EPS-Format	216
C.1 Allgemeines	216
C.2 Überblick	216
C.3 Elemente und Attribute	220
C.4 Stildaten	264
C.5 Anmerkungen zu Festlegungen	285
C.6 Farbnamen	289
C.7 Festlegung des ESP-Formats in der RELAX-NG-Kompakt-Syntax	291
Literaturhinweise	305

	Seite
Bilder	
Bild 1 – Modell für das Erstellen und Verteilen von Inhalten	7
Bild A.1 – Konzeptionelles Diagramm des „Block layout“	19
Bild A.2 – Koordinatensystem.....	20
Bild A.3 – Aufbau des Seitenlayouts.....	21
Bild A.4 – Aufbau des Block layouts	22
Bild A.5 – Attribut bgimagemode	28
Bild A.6 – Attribut layout	29
Bild B.1 – XML-Baumstruktur	79
Bild B.2 – Beispiel mit valign = „middle“	125
Bild B.3 – Beispiel eines nach unten hängenden Großbuchstabens	130
Bild B.4 – Linker und rechter Rand eines Abschnitts	130
Bild B.5 – Horizontale Schreibweise in vertikalem Text	133
Bild B.6 – Ruby	134
Bild B.7 – Beispiel einer Bildaufbereitung für eine Suchseitenobjektinstanz	160
Bild B.8 – Comicobjektinstanz	164
Bild C.1 – Beispiel einer Inhalts-XML-Datei.....	217
Bild C.2 – Beispiel von Stildaten.....	218
Bild C.3 – Beispiel der Markencodierung	219
Bild C.4 – Zahlen, die den Ursprung der absoluten Position auf der Seite abgeben	221
Bild C.5 – Codierungsbeispiel der <section>-Marke	222
Bild C.6 – Verlaufsrichtung für Inhalt/Seite.....	287
Bild C.7 – Beispiel einer Ordner(Verzeichnis)-Struktur	288
Tabellen	
Tabelle A.1 – Datentypen	22
Tabelle A.2 – Attribut von BBeBXylog	23
Tabelle A.3 – Attribut von Info	23
Tabelle A.4 – Attribut von Title.....	24
Tabelle A.5 – Attribut von Author.....	24
Tabelle A.6 – Attribut von Publisher	24
Tabelle A.7 – Attribut von Label.....	25
Tabelle A.8 – Attribut von Cthumbnail	25
Tabelle A.9 – Attribut von TocLabel.....	26
Tabelle A.10 – Attribut von Page	26
Tabelle A.11 – Attribut von Wait	30
Tabelle A.12 – Attribut von TextBlock.....	30
Tabelle A.13 – Attribut rubyalign.....	34
Tabelle A.14 – Attribut rubyadjust.....	34

	Seite
Tabelle A.15 – Attribut rubyoverhang	34
Tabelle A.16 – Attribut empdotsposition	34
Tabelle A.17 – Attribut emplineposition	35
Tabelle A.18 – Attribut emplintype	35
Tabelle A.19 – Attribut align	35
Tabelle A.20 – Attribut blockrule	36
Tabelle A.21 – Attribut von P	38
Tabelle A.22 – Attribut von Plot	38
Tabelle A.23 – Attribut von Fill	39
Tabelle A.24 – Attribut von Space	39
Tabelle A.25 – Attribut von CharButton	39
Tabelle A.26 – Attribut von Gaiji	40
Tabelle A.27 – Attribut von Box	41
Tabelle A.28 – Attribut linetype	41
Tabelle A.29 – Attribut von EmpDots	41
Tabelle A.30 – Attribut von EmpLine	42
Tabelle A.31 – Attribut von DrawChar	43
Tabelle A.32 – Attribut von ImageBlock	43
Tabelle A.33 – Attribut von ButtonBlock	46
Tabelle A.34 – Attribut von BaseButton	48
Tabelle A.35 – Attribut von FocusinButton	48
Tabelle A.36 – Attribut von JumpTo	49
Tabelle A.37 – Attribut von Run	49
Tabelle A.38 – Attribut runoption	49
Tabelle A.39 – Attribut von PushButton	50
Tabelle A.40 – Attribut von UpButton	50
Tabelle A.41 – Attribut von Canvas	50
Tabelle A.42 – Attribut von PutObj	52
Tabelle A.43 – Attribut von Moveto	52
Tabelle A.44 – Attribut von Lineto	53
Tabelle A.45 – Attribut arrowtype	54
Tabelle A.46 – Attribut von DrawBox	54
Tabelle A.47 – Attribut von DrawEllipse	55
Tabelle A.48 – Attribut von RuledLine	56
Tabelle A.49 – Attribut von BookStyle	56
Tabelle A.50 – Attribut von SetDefault	57
Tabelle A.51 – Attribut setwaitprop	58
Tabelle A.52 – Attribut von RegistFont	58

	Seite
Tabelle A.53 – Attribut von BookSetting	59
Tabelle A.54 – Attribut von TextStyle (1 von 2)	60
Tabelle A.55 – Attribut von BlockStyle	64
Tabelle A.56 – Attribut von PageStyle	66
Tabelle A.57 – Attribut von Window	69
Tabelle A.58 – Attribut von PopUpWin	69
Tabelle A.59 – Attribut von Sound	70
Tabelle A.60 – Attribut von SoundStream	70
Tabelle A.61 – Attribut von ImageStream	71
Tabelle A.62 – Attribut von Header	71
Tabelle A.63 – Attribut von Footer	72
Tabelle A.64 – Attribut von eSound	73
Tabelle A.65 – Attribut von Font	73
Tabelle A.66 – Attribut von Image	74
Tabelle A.67 – Attribut von Button	75
Tabelle B.1 – Grundlegende Zeichen für die Lesart	80
Tabelle B.2 – Beispiele für Standardzeichensätze	81
Tabelle B.3 – Für Telefonnummern verwendbare Zeichen	89
Tabelle B.4 – Für E-Mail-Adressen verwendbare Zeichen	89
Tabelle B.5 – Für den Nachschlageschlüssel verwendbare Zeichen	110
Tabelle B.6 – Farbnamen	190
Tabelle B.7 – Beispiele von Standardzeichensätze für verschiedene Sprachen	191
Tabelle B.8 – Beispiel zusätzlicher für die Lesart verwendbarer Zeichen für verschiedene Sprachen	191
Tabelle B.9 – Beispiel für zusätzliche Sortierregeln	192
Tabelle B.10 – Beispiele zusätzlicher sprachspezifischer Attribute für <enable_key_type>	192
Tabelle C.1 – Struktur des ESP-Formats	217
Tabelle C.2 – Nutzbare Zeilenumbruchcodes	218
Tabelle C.3 – Symbole halber Breite für Dateinamen	219
Tabelle C.4 – Schreibweisen	220
Tabelle C.5 – Entsprechung Eigenschaft – Selektor	278
Tabelle C.6 – Marken, die zu unabhängigen Blöcken führen	286
Tabelle C.7 – Eigenschaften für das Parsen	286
Tabelle C.8 – Ein Beispiel zur Umsetzung auf numerische Werte	289
Tabelle C.9 – Farbnamen	290

Einleitung

Der Markt für E-Books und E-Publishing fordert eine Normung der Formate für den E-Book-Datenaustausch zwischen den beteiligten Stellen: Autoren, Datenaufbereitern, Verlegern (Publisher) und Lesegeräten (E-Book-Reader). Die Formate werden in Eingabeformate, allgemeine Formate und Leserformate eingeteilt. Das einzureichende Eingabeformat muss die Interaktion zwischen Autoren und Datenaufbereitern ermöglichen. Das Leserformat hängt vom Lesegerät ab. Das allgemeine Format muss ein Format für den Datenaustausch zwischen Datenaufbereiter und Publisher sein und sollte deshalb von E-Publishing-Geräten unabhängig sein.

1 Anwendungsbereich

In dieser Internationalen Norm wird ein allgemeines Format für Multimedia-E-Publishing für den Datenaustausch zwischen Datenaufbereitern und Verlegern (Publisher) für E-Books festgelegt. Das Format muss eine Reihe von Anforderungen des Verlegers erfüllen: Überprüfbarkeit, Erweiterbarkeit und eine heterogene logische Struktur.

2 Normative Verweisungen

Die folgenden zitierten Dokumente sind für die Anwendung dieses Dokuments erforderlich. Bei datierten Verweisungen gilt nur die in Bezug genommene Ausgabe. Bei undatierten Verweisungen gilt die letzte Ausgabe des in Bezug genommenen Dokuments (einschließlich aller Änderungen).

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

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

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

ISO 646, *Information technology – ISO 7-bit-coded character set for information interchange*

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

ISO 8601, *Data elements and interchange formats – Information interchange – Representation of dates and times*

3 Begriffe

Für die Anwendung dieses Dokuments gelten die folgenden Begriffe.

3.1

Multimedia-E-Book

Multimedialinhalt, bestehend aus Texten, Grafiken, Ton und/oder Bild

[IEC 62229]

3.2

Datenaufbereiter

Organisation oder Person, die ein E-Book ausarbeitet. Ein Beispiel für den Datenaufbereiter ist der Editor.

[IEC 62229]

3.3

Publisher, Verleger

eine Organisation oder Person, die ein E-Book herausgibt und verteilt

[IEC 62229]

3.4

allgemeines Format

Format für Multimedia-E-Book-Inhalte, das durch den Datenaufbereiter erstellt und aufbereitet und durch den Publisher zur Verteilung als E-Book in ein Reader-Format gebracht wird

4 Stellung und Anforderungen für das allgemeine Format

4.1 Allgemeines Format des Modells für das Erstellen und Verteilen von Inhalten

Im konzeptionellen Modell für Multimedia-E-Publishing [IEC 62229] wird, wie in Bild 1 gezeigt, ein Modell für das Erstellen und Verteilen von Inhalten definiert.

Autor ← (1) → Datenaufbereiter ← (2) → Publisher ← (3) → Leser

Bild 1 – Modell für das Erstellen und Verteilen von Inhalten

Zwischen den benachbarten Verarbeitungsstufen Datenaufbereitung und Publishing findet der Datenaustausch der E-Book-Inhalte unter Anwendung des allgemeinen Formates statt. Zusammenstellungen von Originaltexten, Grafiken, Ton und/oder Bild liefert der Autor, E-Book-Daten erstellt und bereitet der Datenaufbereiter in ein allgemeines Format auf. Die E-Book-Daten werden gespeichert und zu dem Publisher weitergeleitet.

Die Publisher modifizieren das allgemeine Format in Reader-Formate, die für die Verteilungs-Pläne und Verteilungs-Einrichtungen geeignet sind. Das allgemeine Format kann auch für die Verteilung der E-Books benutzt werden.

4.2 Anforderungen für das allgemeine Format

Das allgemeine Format kann die folgenden Anforderungen der Datenaufbereiter und Publisher erfüllen:

a) Logische Struktur:

Das allgemeine Format muss eine logische Struktur aufweisen, die leicht abgeändert werden kann.

b) Style-Festlegung:

Die logischen Elemente in dem allgemeinen Format sollten entsprechend einer Style-Festlegung für das Erstellen eines geeigneten Formates für den E-Book Reader geschaffen werden.

c) Heterogene Strukturen:

Multimedia-E-Books dürfen der Art der Inhalte entsprechend verschiedenartige Strukturen wie Mathematik, Chemie und Musiknoten enthalten. Diese spezifischen Strukturen sollten in das allgemeine Format importiert werden.

d) Metadaten:

Das allgemeine Format sollte zur Verwaltung der Daten der E-Book-Inhalte einige Metadaten unterstützen.

5 Notation

Die logische Struktur des allgemeinen Formates wird unter Verwendung von RELAX NG-Schema in ISO/IEC 19757-2 und seiner Änderung 1 beschrieben.

6 Logische Struktur

Die logische Struktur des allgemeinen Formates: ebook-g liefert ein einfaches und erweiterbares Format ebook-g-core sowie bestehende und aktuell verwendete Formate – BBeB Xylog, XMDF- und ESP-Formate. Die Strukturen von BBeB Xylog-, XMDF- und ESP-Formaten sind in Anhang A, Anhang B und Anhang C dargestellt.

```
default namespace="http://tc100.iec.ch/2005/ebook/generic"
start = ebook-g
ebook-g = ebook-g-core
```

```
| external "bbehxylog.rnc"
| external "xmdf.rnc"
| external "esp.rnc"
ebook-g-core = meta-g & body-g
meta-g = external "meta-g.rnc"
body-g = element body-g { body }
body =
    title*,
    foreword*,
    introduction*,
    titled-clause+,
    annex*,
    bibliography*
title = element title {
    element main { text },
    element sub { text }?
}
foreword = element foreword { text }
introduction = element introduction { block+ }
titled clause = element clause { id, titel, 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 }
reference-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 Semantiken

Elemente von ebook-g-core sollten den zutreffenden Style-Festlegungen entsprechend wiedergegeben werden. Aktuelle Style-Festlegungen durch Style-Sprachen, z. B. XSL oder DSSSL, liegen außerhalb des Anwendungsbereiches dieser Internationalen Norm.

Die Inhalte von meta-g.rnc, table.rnc und figure.rnc sind nicht innerhalb einer Architektur von ebook-g-core festgelegt.

Anhang A (normativ)

BBeB Xylog-Format

A.1 Allgemeines

In diesem Anhang ist das in Abschnitt 6 der Beschreibung gezeigte BBeB Xylog-Format festgelegt.

Dieses Format ist zur Verwendung durch zwei Benutzergruppen vorgesehen: Hersteller von E-Book-Inhalten und Entwickler entsprechender Werkzeuge. Die Hersteller der Inhalte können diese Festlegung als allgemeines Format für die Speicherung der Daten von Inhalten benutzen, die auf den Festlegungen der Internationalen Norm beruhen. Der entsprechende Werkzeugentwickler kann dieses Format als Bezug benutzen, um die Spezifikation für die Anzeigegeräte festzulegen und um die Konvertierungsregeln zwischen dieser Festlegung und den anderen Datenarten wie XML oder HTML zu definieren.

Die Besonderheiten der in diesem Anhang enthaltenen Festlegungen sind folgende:

- a) Der „Eindruck eines Buches aus Papier“ durch ein Layout mit zwei Bildschirmen und modifizierten Zeichen usw.
- b) Der „digitale Eindruck“ durch Ton, Springen zu Seiten und interaktive Aktionen usw.

Die für die oben behandelten Eindrücke und ihre problemlose Bedienung notwendigen Elemente werden in diesen Festlegungen definiert.

A.2 Beschreibung der Struktur

```
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.Title, 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.CommonObjects
  I element.BlockSpace
  I element.RuledLine
  I 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 I 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" I "baseline" I "top" I "bottom"? }
element.CR = element CR { empty }
element.Fill =
    element Fill {
        attribute code { text }
        empty
    }
element.Space = element Space { attlist.xsize, empty }
element.CharButton = element CharButton { attlist.refid, SimpleChar1 }
element.Rubi = element Rubi { (element.Oyamoji, element.Rubimoji)+ }
element.Oyamoji = element Oyamoji { SimpleChar0 }
element.Gaiji = element Gaiji { attlist.Gaiji, text }
attlist.Gaiji &=
    attlist.refid,
    attribute fontfacename { text }?,
    attribute code { text }
element.AltString = element AltString { element.Org, element.Alt }
element.Org = element Org { text }
element.Alt = element Alt { text }
element.Rubimoji = element Rubimoji { SimpleChar0 }
element.Box = element Box { attlist.Box, SimpleChar0 }
attlist.linetype &=
    attribute linetype { "solid" | "dotted" | "dashed" | "double" }
attlist.Box &= attlist.linetype?
element.EmpDots = element EmpDots { attlist.EmpDots, SimpleChar0 }
attlist.Empdotsposition &=
    attribute empdotsposition { "before" | "after" }
attlist.EmpdotsChar &=
    attribute empdotscode { text },
    attribute empdotsfontname { text },
    attribute refempdotsfont { text }
attlist.EmpDots &= attlist.Empdotsposition ?, attlist.EmpdotsChar ?
element.Empline = element Empline { attlist.Empline, SimpleChar0 }
attlist.Emplineposition &=
    attribute emplineposition { "before" | "after" }
attlist.Emplinetype &=
    attribute emplinetype {
        "none" | "solid" | "dotted" | "dashed" | "double"
    }
attlist.Empline &= attlist.Emplinetype?, attlist.Emplineposition?
element.Sub = element Sub { SimpleChar0 }
element.Sub = element Sub { 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 Image Block { 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 Button Block {
    attlist.Button Block,
    element.BaseButton?,
    element.FocusinButton?,
    element.PushButton?,
    (element.UpButton?)
  >> sch:pattern [
    name = "Style Check"
    "\x{a}"~
    "
  ]
  sch:rule [
    context = "ButtonBlock"
    "\x{a}"~
    "
  ]
  sch:assert [
    test = "@blockstyle = //Style/BlockStyle/@stylelabel"
    "No BlockStyle"
  ]
  "\x{a}"~
  "
  ]
  "\x{a}"~
  "
  ]
}
attlist.ButtonBlock &=
  attlist.id,
  attribute blockstyle { text },
  attlist.Common.Block
attlist.refimage &= attribute refimage { xsd:string }
element.BaseButton = element BaseButton { attlist.refimage?, empty }
element.FocusinButton =
  element FocusinButton {
    attlist.refimage?,
    (element.JumpTo
    | element.Run
    | element.SoundStop
    | element.CloseWindow)*
  }
element.JumpTo = element JumpTo { attlist.jump, empty }
attlist.jump &=
```

```
        attribute reffpage { xsd:string },
        attlist.refid
element.Run = element Run { attlist.Run, empty }
attlist.Run &=
    [ a:defaultValue = "normal" ]
    attribute runoption {
        "normal" | "opposite" | "center" | "opposite-center"
    }?,
    attlist.refid
element.SoundStop = element SoundStop { empty }
element.CloseWindow = element CloseWindow { empty }
element.PushButton =
    element PushButton {
        attlist.refimage?,
        (element.JumpTo
        | element.Run
        | element.SoundStop
        | element.CloseWindow)*
    }
element.UpButton =
    element UpButton {
        attlist.refimage?,
        (element.JumpTo
        | element.Run
        | element.SoundStop
        | element.CloseWindow)*
    }
element.BlockSpace = element BlockSpace { attlist.BlockSpace, empty }
attlist.BlockSpace &=
    attribute xspace { xsd:short }?,
    attribute yspace { xsd:short }?
element.Canvas =
    element Canvas {
        attlist.Canvas, (element.Common.Canvas | element.Wait)*
    }
element.Common.Canvas =
    element.PutObj
    | element.Moveto
    | element.Lineto
    | element.DrawBox
    | element.DrawEllipse
attlist.Canvas &=
    attribute canvaswidth { xsd:unsignedShort },
    attribute canvasheight { xsd:unsignedShort },
    attribute blockrule { "block-fixed" | "block-adjustable" }?,
    attlist.id,
    attlist.Common.Canvas
attlist.Layout &= attribute layout { "LrTb" | "TbRl" }
attlist.Common.Frame &=
    attribute framewidth { xsd:unsignedShort }?,
    attribute framecolor { text }?,
    attribute framemode { "curve" | "square" }?
attlist.Common.Canvas &=
    attribute bgcolor { text }?,
    attlist.Layout?,
    attlist.Common.Frame
element.PutObj = element PutObj { attlist.PutObj, empty }
attlist.x1.unsigned &= attribute x1 { xsd:unsignedShort }
attlist.x1.signed &= attribute x1 { xsd:short }
attlist.y1.unsigned &= attribute y1 { xsd:unsignedShort }
attlist.y1.signed &= attribute y1 { xsd:short }
attlist.PutObj &=
    attlist.x1.unsigned, attlist.y1.unsigned, attlist.refid
element.Wait =
    element Wait {
        attribute time { xsd:short },
        empty
```

```
    }
element.Moveto =
    element Moveto { attlist.x1.signed, attlist.y1.signed, empty }
element.Lineto = element Lineto { attlist.Lineto, empty }
attlist.Common.line &=
    attribute linewidth { xsd:unsignedShort }?,
    attribute linecolor { text }?,
    attlist.linetype?
attlist.Lineto &=
    attlist.x1.signed,
    attlist.y1.signed,
    attlist.Common.line,
    attribute arrowtype { text}?
element.DrawBox = element DrawBox { attlist.DrawBox, empty }
attlist.DrawBox &=
    attlist.x1.signed,
    attlist.y1.signed,
    attlist.Common.line,
    attribute fillcolor { text}?
element.DrawEllipse = element DrawEllipse { attlist.DrawEllipse, empty }
attlist.DrawEllipse &=
    attlist.x1.signed,
    attlist.y1.signed,
    attlist.Common.line,
    attribute fillcolor { text}?
element.RuledLine = element RuledLine { attlist.RuledLine, empty }
attlist.RuledLine &=
    attlist.linetype,
    attribute linewidth { xsd:unsignedShort },
    attribute linelength { xsd:unsignedShort },
    attribute linecolor { text }?
element.Style =
    element Style {
        element.BookStyle,
        (element.TextStyle | element.BlockStyle | element.PageStyle)*
    }
element.BookStyle =
    element BookStyle {
        attlist.Style,
        element.SetDefault?,
        element.RegistFont*,
        element.BookSetting
    }
attlist.Style &=
    attribute stylelabel { text }
    attlist.id
element.SetDefault = element SetDefault { attlist.SetDefault, empty }
attlist.Rubyoverhang &= attribute rubyoverhang { "auto" | "none" }
attlist.Rubyalign &= attribute rubyalign { "start" | "center" }
attlist.Rubyadjust &= attribute rubyadjust { "line-edge" | "none" }
attlist.Setwaitprop &= attribute setwaitprop { "replay" | "noreplay" }
attlist.SetDefault &=
    attlist.Rubyalign,
    attlist.Rubyoverhang,
    attlist.Emphdotsposition,
    attlist.Emplineposition,
    attlist.Emplintype,
    attlist.EmphdotsChar,
    attlist.Rubyadjust,
    attlist.Setwaitprop
element.RegistFont = element RegistFont { attlist.RegistFont, empty }
attlist.FontEncoding &= attribute encoding { "TTF" | "OTF" | "BF" }
attlist.RegistFont &=
    attribute fontfilename { text },
    attribute file { text },
    attribute fontname { text },
    attlist.FontEncoding
```

```
element.BookSetting = element BookSetting { attlist.BookSetting, empty }
attlist.BookSetting &=
    attribute bindingdirection { "Lr" | "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" | "title" | "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" I "lower" I "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

Das Format „BBeB Book“ verwendet das „Block layout“-Konzept. Im „Block layout“ wird der anzuzeigende Inhalt in einem „Block“ genannten Bereich angezeigt. Das Layout setzt sich aus mehreren „Blöcken“ zusammen. Jeder „Block“ wird an einer entsprechenden Stelle angeordnet. Das „BBeB XML Xylog“-Dateiformat definiert die Implementierungsregeln, die auf diesem „Block layout“-Konzept beruhen (siehe Bild A.1).

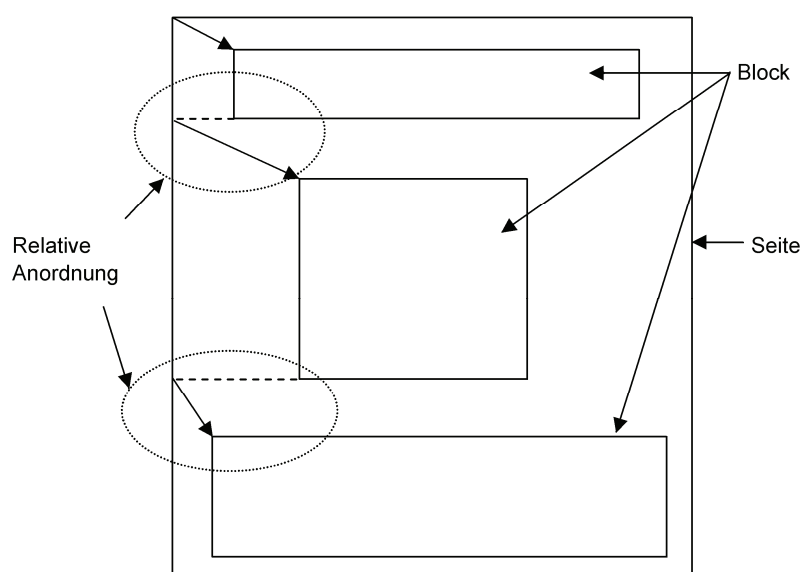


Bild A.1 – Konzeptionelles Diagramm des „Block layout“

A.3.2 Block

In dieser Festlegung werden die Elemente **TextBlock**, **ImageBlock**, **ButtonBlock** und **Canvas** definiert. Sie sind auf einer „Page“ als ein „Block“ angeordnet. Sie werden als Unterelement des Elementes **Page** beschrieben, zusammen mit dem Element **BlockSpace**, das die relative Lage der Anordnung anzeigt.

Es ist möglich, im „Block“ ein Hintergrundbild, eine Hintergrundfarbe und Rahmen festzulegen.

A.3.3 Koordinatensystem

Diese Norm verwendet ein zweidimensionales Koordinatensystem (X, Y), das auf dem vertikalen oder horizontalen Layout beruht (siehe Bild A.2).

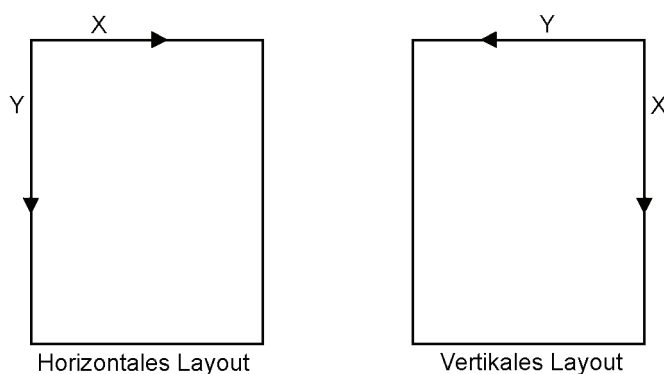


Bild A.2 – Koordinatensystem

A.4 Terminologie

A.4.1 Seitenlayout

Für das Seitenlayout werden die folgenden Begriffe definiert (siehe Bild A.3):

Bildschirmgröße (en: screen size)

Bildschirmgröße des Seiteninhalts, der mit einer speziellen Anzeigeeinrichtung betrachtet wird und durch die Werte der Attribute **screenwidth** und **screenheight** in dem Element **BookSetting** definiert ist (siehe A.5.71)

Oberrand (en: topmargin)

der Abstand zwischen der oberen Kante des Bildschirms und dem Kopfbereich

Kopfbereich (en: header area)

Bereich, in dem der/die Header angeordnet ist/sind

Kopfhöhe (en: headheight)

die Höhe des Kopfbereiches

Kopfzwischenraum (en: headsep)

der Abstand zwischen Kopfbereich und Haupt-Textbereich

Haupt-Textbereich (en: main text area)

Bereich, in dem der Haupt-Inhalt angeordnet ist

Texthöhe (en: textheight)

die Höhe des Haupt-Textbereiches

Textbreite (en: textwidth)

die Breite des Kopfbereiches, des Haupt-Textbereiches und des Fußbereiches

Fußabstand Hauptbereich – Fußbereich (en: footspace)
der Abstand zwischen dem unteren Ende des Haupt-Textbereiches und der Unterkante des Fußbereiches

Fußhöhe (en: footheight)
die Höhe des Fußbereiches

ungeradseitiger linker Seitenrand (en: oddsidemargin)
der Rand zwischen der linken Kante des Bildschirms und der linken Kante des Haupt-Textbereiches auf den ungeraden Buchseiten

geradseitiger linker Seitenrand (en: evensidemargin)
der Rand zwischen der linken Kante des Bildschirms und der linken Kante des Haupt-Textbereiches auf den geraden Buchseiten

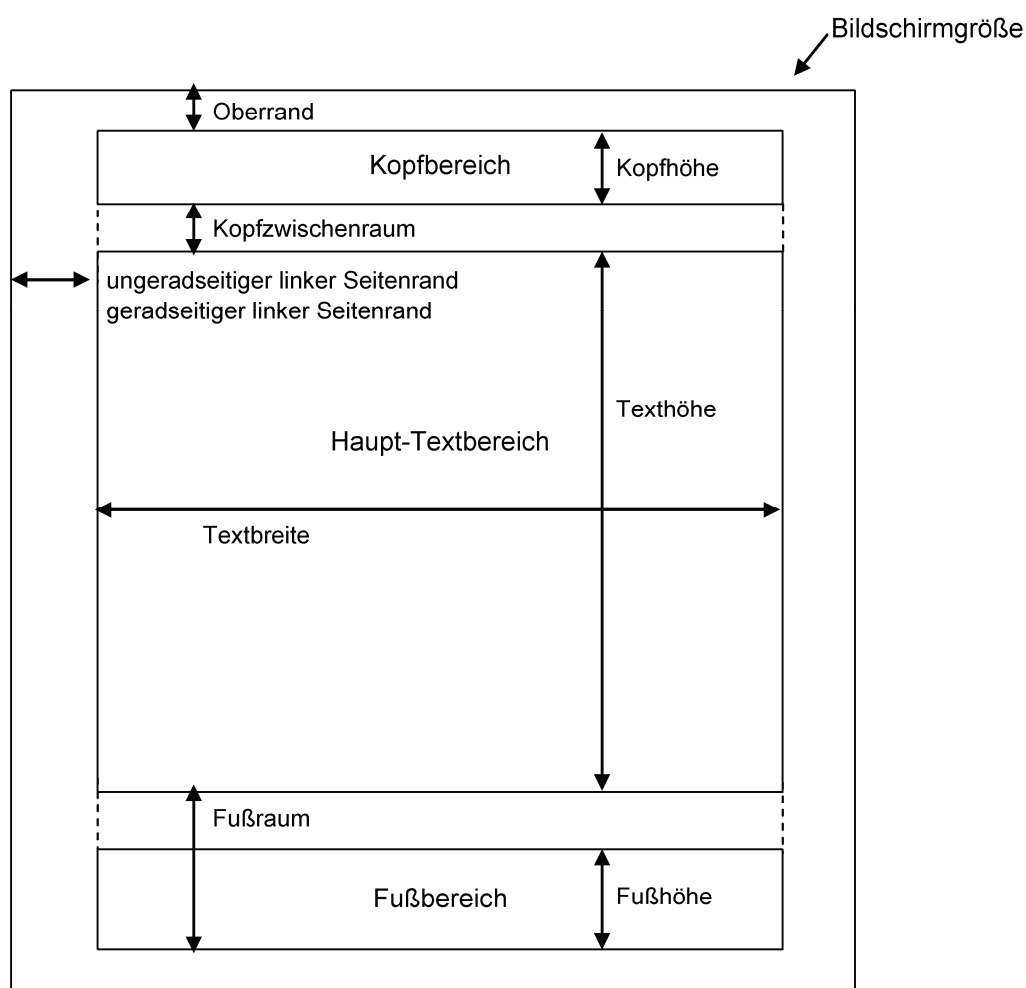


Bild A.3 – Aufbau des Seitenlayouts

A.4.2 Block layout

Für den Bereich des Block layouts werden die folgenden Begriffe definiert (siehe Bild A.4):

Wiedergabebereich (en: rendering area)
in einem Block definierter Bereich, in dem Text oder Bild platziert wird

Wiedergabebereich-Oberrand (en: topskip)
Rand zwischen der Oberkante des Blocks und der Oberkante des Wiedergabebereiches für beide Seiten

Wiedergabebereich-Seitenrand (en: sidemargin)

Rand zwischen der Seitenkante eines Blocks und der Seitenkante des Wiedergabebereiches

Wiedergabebereich-Unterrand (en: footskip)

Rand zwischen der Unterkante des Blocks und der Unterkante des Wiedergabebereiches

Blockhöhe (en: blockheight)

Höhe des „Blocks“

Blockbreite (en: blockwidth)

Breite des „Blocks“

Zeileneinzug (en: parindent)

Zeileneinzug des Paragraphen, gemessen von der Seitenkante des Wiedergabebereiches

Absatzabstand (en: parskip)

Abstand zwischen der letzten Zeile eines Absatzes und der ersten Zeile des nächsten Absatzes

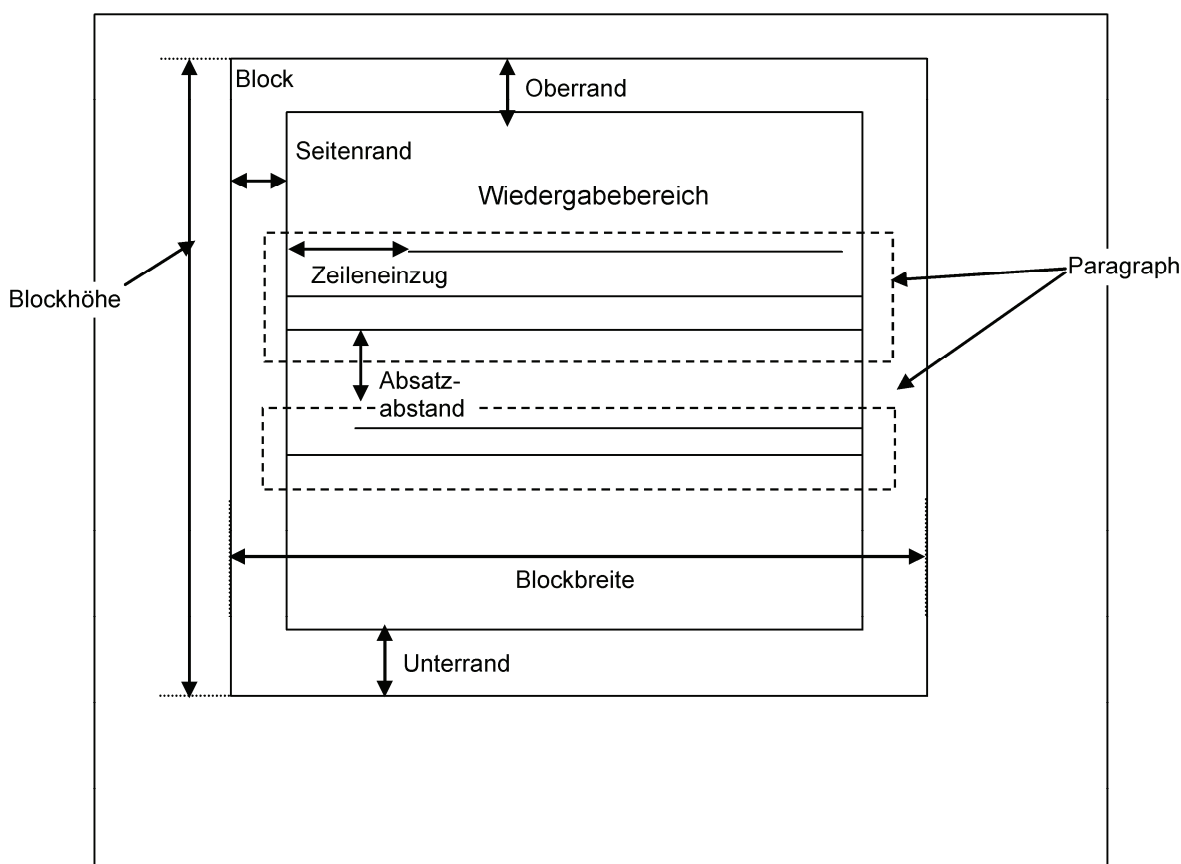


Bild A.4 – Aufbau des Block layouts

A.4.3 Datentypen

Die folgenden Typdefinitionen sind in dieser Norm als Werte der XML-„Attribute“ definiert.

Tabelle A.1 – Datentypen (1 von 2)

Typendefinition	Bedeutung	Beispiel
String = Zeichenkette	Legt eine übliche Zeichenkette fest.	„after“, „solid“
String for sorting = Sortierfolge	Nur für die in „Character Set for Sorting Specification“ definierte Zeichenfolge.	„Kennedy John F.“

Tabelle A.1 – Datentypen (2 von 2)

Typendefinition	Bedeutung	Beispiel
File path = Dateipfad	Gibt den Pfad an, in dem die Datei angeordnet ist.	„D:\sample.jpg“, „..\data\white.mp3“
Decimal string = Dezimal-Zeichenkette	Gibt eine Dezimalzahl als Zeichenkette an.	„10“, „20“
Hexadecimal string = Hexadezimal-Zeichenkette	Gibt eine Hexadezimalzahl als Zeichenkette an. Am Anfang der Zahl steht „0x“, um zu zeigen, dass die folgende Zahl hexadezimal ist.	„0x12“, „0x3333“
Style string = Format-Zeichenkette	Legt eine Zeichenkette fest, die als Style-Kennzeichnung verwendet wird.	„Main Block“, „Cover Page“, „Section Title Text“
Object ID string = Objekt-Kennzeichnungsfolge	Legt eine Zeichenkette fest, die als Objekt-ID (Objekt-Bezeichner) verwendet wird.	„33“, „Object12“
COLORREF string = COLORREF-Zeichenkette	Legt eine hexadezimale 32-Bit-Zeichenkette fest, die die Farbe signalisiert. Das Beschreibungsformat ist 0xaaarrgbb. (aa: alpha-Kanal [in dieser Spezifikation als „00“ oder „ff“ festgelegt], „rr“: roter Wert, „gg“: grüner Wert, „bb“: blauer Wert).	„0x00ff0000“ (red)

A.5 Elemente und Attribute

A.5.1 BBeBXylog

Das Element **BBeBXylog** zeigt den Bereich des gesamten Inhalts im Dateiformat „BBeB XML Xylog“ an.

Tabelle A.2 – Attribut von BBeBXylog

Attributname	Vorgabewert	Datentyp	Bemerkungen
version (erforderlich)		Zeichenkette	Definiert als „1,0“.

Das Attribut **version** gibt die entsprechende Versionsnummer für das Dateiformat „BBeB XML Xylog“ an. Nach dieser Norm muss der Wert „1,0“ sein.

A.5.2 BookInformation

Das Element **BookInformation** gibt bibliografische Informationen und Informationen über die TOC (en: Table Of Content).

A.5.3 Info

Das Element **Info** gibt bibliografische Informationen und zusätzliche Informationen über den Inhalt.

Tabelle A.3 – Attribut von Info

Attributname	Vorgabewert	Datentyp	Bemerkungen
version (erforderlich)		Zeichenkette	Definiert als „1,0“.

Das Attribut **version** gibt die Versionsnummer des Formates der bibliografischen Informationen an. Nach dieser Norm muss der Wert „1,0“ sein.

A.5.4 BookInfo

Das Element **BookInfo** gibt bibliografische Informationen des Inhalts.

A.5.5 Title

Das Element **Title** gibt den Titel des Inhalts an.

Tabelle A.4 – Attribut von Title

Attributname	Vorgabewert	Datentyp	Bemerkungen
reading (erforderlich)		Sortierfolge	

Das Attribut **reading** gibt die Sortierfolge des Inhalts nach Titeln an. Im Japanischen können die „Japanese syllabary character“ usw. benutzt werden. Im Englischen kann das „Alphabet“ usw. benutzt werden. Die Einzelheiten der in diesem Attribut verwendeten Zeichensätze dürfen in gegenseitiger Absprache festgelegt werden.

A.5.6 Author

Das Element **Author** gibt den Namen des Autors des Inhalts an. Wenn Benutzer mehrere Autorennamen angeben wollen, müssen die Namen seriell in einem Element angegeben werden.

Tabelle A.5 – Attribut von Author

Attributname	Vorgabewert	Datentyp	Bemerkungen
reading (erforderlich)		Sortierfolge	

Das Attribut **reading** gibt die Sortierfolge des Inhalts nach Autoren an. Im Japanischen können die „Japanese syllabary character“ usw. benutzt werden. Im Englischen kann das „Alphabet“ usw. benutzt werden. Die Einzelheiten der in diesem Attribut verwendeten Zeichensätze dürfen in gegenseitiger Absprache festgelegt werden.

A.5.7 BookID

Das Element **BookID** gibt die ID des Inhalts an. Diese ID dient zur Unterscheidung von „privatem Inhalt“ und „öffentlichem Inhalt“. Die BookID für privaten Inhalt muss durch die Zeichen „FB“ und folgende Zeichen mit 14 Stellen oder weniger (insgesamt 16 Stellen oder weniger) angegeben werden. Die BookID für öffentlichen Inhalt darf in gegenseitiger Absprache festgelegt werden.

A.5.8 Publisher

Das Element **Publisher** gibt den Namen des E-Book-Publishers des Inhalts an.

Tabelle A.6 – Attribut von Publisher

Attributname	Vorgabewert	Datentyp	Bemerkungen
reading (erforderlich)		Sortierfolge	

Das Attribut **reading** gibt die Sortierfolge des Inhalts durch den Publisher an. Im Japanischen können die „Japanese syllabary character“ usw. benutzt werden. Im Englischen kann das „Alphabet“ usw. benutzt werden. Die Einzelheiten der in diesem Attribut verwendeten Zeichensätze dürfen in gegenseitiger Absprache festgelegt werden.

A.5.9 Label

Das Element **Label** gibt den Namen des Bezeichners an.

Tabelle A.7 – Attribut von Label

Attributname	Vorgabewert	Datentyp	Bemerkungen
reading (erforderlich)		Sortierfolge	

Das Attribut **reading** gibt die Sortierfolge des Inhalts nach Bezeichnern an. Im Japanischen können die „Japanese syllabary character“ usw. benutzt werden. Im Englischen kann das „Alphabet“ usw. benutzt werden. Die Einzelheiten der in diesem Attribut verwendeten Zeichensätze dürfen in gegenseitiger Absprache festgelegt werden.

A.5.10 Category

Das Element **Category** gibt die Art des Inhalts an. Die Anzahl dieses Elementes darf maximal zwei sein.

A.5.11 Classification

Das Element **Classification** informiert darüber, welche Art von Daten im Inhalt enthalten sind (z. B. Ton, Farbbilder).

A.5.12 FreeText

Das Element **FreeText** gibt eine formlose Beschreibung des Inhalts (z. B. Inhaltsübersicht).

A.5.13 DocInfo

Das Element **DocInfo** gibt Informationen über Miniaturbilder, Sprache usw.

A.5.14 Cthumbnail

Das Element **Cthumbnail** gibt den Dateinamen des Miniaturbildes für den Inhalt an.

Tabelle A.8 – Attribut von Cthumbnail

Attributname	Vorgabewert	Datentyp	Bemerkungen
file (erforderlich)		Dateipfad	Beschreibt den absoluten Pfad oder den möglichen relativen Pfad der Datei, der für das Miniaturbild benutzt werden muss.

Das Attribut **file** gibt den Dateinamen und den Pfad der Miniaturbild-Datei an.

A.5.15 Language

Das Element **Language** gibt die im Inhalt verwendete Hauptsprache an. Es sollten die „ISO 639 language codes“ benutzt werden, z. B. Japanese: „Ja“.

A.5.16 Creator

Das Element **Creator** gibt den Namen des Verfassers oder des Studios an, das den Inhalt erstellt hat.

A.5.17 SumPage

Das Element **SumPage** gibt die Anzahl von „Views“ (sichtbare Seiten bei den Bedingungen des Elementes **BookSetting**) des Inhalts an.

A.5.18 Keyword

Das Element **Keyword** gibt Schlüsselworte für das Suchen des Inhalts an.

A.5.19 TOC

Das Element **TOC** beschreibt das tabellarische Inhaltsverzeichnis.

A.5.20 TocLabel

Das Element **TocLabel** gibt die Zeichenkette an, die als tabellarisches Inhaltsverzeichnis angezeigt wird.

Tabelle A.9 – Attribut von TocLabel

Attributname	Vorgabewert	Datentyp	Bemerkungen
refobj (erforderlich)		Objekt-ID-Zeichenkette	Gibt die „objid“ der Elemente TextBlock , ImageBlock , ButtonBlock und Canvas an.
refpage (erforderlich)		Objekt-ID-Zeichenkette	Gibt die „objid“ des Elementes Page an.

A.5.20.1 refobj

Das Attribut **refobj** gibt die „objid“ des als das Sprungziel angegebenen Elementes an. Das als „objid“ angegebene Element sollte in dem Element **Page** angeordnet werden, das als „refpage“ angegeben wird.

A.5.20.2 refpage

Das Attribut **refpage** gibt die „objid“ des Elementes **Page** an, einschließlich des Elementes, das als Sprungziel angegeben ist.

A.5.21 Main

Das Element **Main** beschreibt den „Main content“.

A.5.22 Page

Im Element **Page** werden die Elemente beschrieben, die auf der „Page“ zusammengestellt sind, und ihre Layoutangaben. Für die Attribute des Elementes **Page** wird empfohlen, nur sich von den Informationen des Elementes **PageStyle** (durch „pagestyle“ festgelegt) unterscheidende Informationen anzugeben.

Tabelle A.10 – Attribut von Page (1 von 2)

Attributname	Vorgabewert	Datentyp	Bemerkungen
bgimagemode		Zeichenkette	Gibt „fix“, „tile“ oder „centering“ an.
evenfooterid		Objektkennungsfolge	Gibt die „objid“ des Elementes Footer an.
evenheaderid		Objektkennungsfolge	Gibt die „objid“ des Elementes Header an.

Tabelle A.10 – Attribut von Page (2 von 2)

Attributname	Vorgabewert	Datentyp	Bemerkungen
evensidemargin	(Ganzzahl ohne Vorzeichen)	Dezimale Zeichenkette	[Pixel]
footheight	(Ganzzahl mit Vorzeichen)	Dezimale Zeichenkette	[Pixel]
footspace	(Ganzzahl mit Vorzeichen)	Dezimale Zeichenkette	[Pixel]
headheight	(Ganzzahl mit Vorzeichen)	Dezimale Zeichenkette	[Pixel]
headsep	(Ganzzahl mit Vorzeichen)	Dezimale Zeichenkette	[Pixel]
layout		Zeichenkette	Gibt „LrTb“ oder „TbRl“ an.
objid (erforderlich)		Zeichenkette	Gibt die eindeutige Zeichenkette der Datei an.
oddfooterid		Dezimale Zeichenkette	Gibt die „objid“ des Elementes Footer an.
oddheaderid		Dezimale Zeichenkette	Gibt die „objid“ des Elementes Header an.
oddsidemargin	(Ganzzahl ohne Vorzeichen)	Dezimale Zeichenkette	[Pixel]
pageposition		Zeichenkette	Gibt „upper“, „lower“ oder „any“ an.
pagestyle (erforderlich)		Style-Zeichenkette	Es ist erforderlich, die angegebene Style-Zeichenfolge auf „stylelabel“ des bezeichneten Elementes PageStyle abzustimmen.
refbgimage		Objekt-ID-Zeichenkette	Gibt die „objid“ des Elementes Image an. Außerdem ist es nötig, gleichzeitig das Attribut bgimagemode anzugeben.
textheight		Dezimale Zeichenkette	[Pixel]
textwidth		Dezimale Zeichenkette	[Pixel]
topmargin		Dezimale Zeichenkette	[Pixel]

A.5.22.1 pagestyle

Das Attribut **pagestyle** gibt das „stylelabel“ des Elementes **PageStyle** mit der notwendigen Information für das „Page“-Layout an.

A.5.22.2 refbgimage

Das Attribut **refbgimage** gibt die „objid“ des Elementes **Image** an, das als Hintergrund des Elementes **Page** gezeigt wird. Wenn die „objid“ nicht angegeben ist, wird das Bild nicht angezeigt.

A.5.22.3 pageposition

Das Attribut **pageposition** gibt an, ob die Anfangsposition die linke oder die rechte Seite des Anzeigegerätes mit zwei Bildschirmen ist.

„upper“ wird angegeben, wenn sie auf der oberen Seite angezeigt wird, „lower“ wird angegeben, wenn sie auf der unteren Seite gezeigt wird, und „any“, wenn sie nach der vorhergehenden Seite angezeigt wird.

Das Attribut **bindingdirektion** „Rl“ (meistens vertikal fließender Inhalt) in dem Element **BookSetting**:
„upper“ bedeutet die Anfangsseite ist rechts,
„lower“ bedeutet die Anfangsseite ist links,
„any“ bedeutet Anzeige fortlaufend von der vorhergehenden Seite.

Das Attribut **bindingdirektion** „Lr“ (meistens horizontal fließender Inhalt) in dem Element **BookSetting**:
„upper“ bedeutet die Anfangsseite ist links,
„lower“ bedeutet die Anfangsseite ist rechts,
„any“ bedeutet Anzeige fortlaufend von der vorhergehenden Seite.

A.5.22.4 oddheaderid

Das Attribut **oddheaderid** gibt die „objid“ des Elementes **Header** an, das auf der ungeraden „Page“ (obere Seite) der beiden Bildschirmanzeigen angezeigt wird.

A.5.22.5 evenheaderid

Das Attribut **evenheaderid** gibt die „objid“ des Elementes **Header** an, das auf der geraden „Page“ (untere Seite) der beiden Bildschirmanzeigen angezeigt wird.

A.5.22.6 oddfooterid

Das Attribut **oddheaderid** gibt die „objid“ des Elementes **Footer** an, das auf der ungeraden „Page“ (obere Seite) der beiden Bildschirmanzeigen angezeigt wird.

A.5.22.7 evenfooterid

Das Attribut **evenfooterid** gibt die „objid“ des Elementes **Footer** an, das auf der geraden „Page“ (untere Seite) der beiden Bildschirmanzeigen angezeigt wird.

A.5.22.8 bgimagemode

Das Attribut **bgimagemode** gibt an, wie das im Attribut **refbgimage** angegebene „Image“ angeordnet werden sollte.

Der Wert ist „fix“: Die Bildposition liegt am Ausgangspunkt der „main text area“.

Der Wert ist „tile“: Das Bild füllt die „main text area“ vom Anfangspunkt an aus.

Der Wert ist „center“: Die Bildposition liegt in der Mitte der „main text area“.

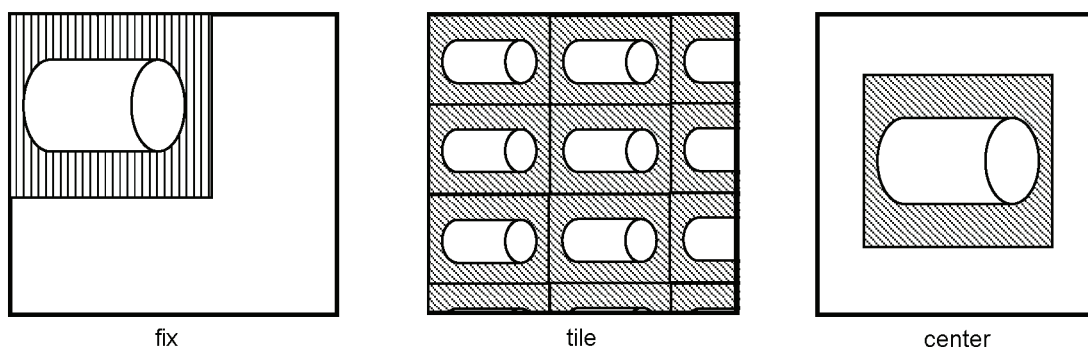


Bild A.5 – Attribut bgimagemode

A.5.22.9 evensidemargin

Das Attribut **evensidemargin** stellt bei einem Zweibildschirm-Anzeigegerät den linken Rand der „Page layout composition“ ein: im horizontalen Layout auf der linken Seite und im vertikalen Layout auf der rechten Seite.

A.5.22.10 footheight

Das Attribut **footheight** stellt die „footheight“ der „Page layout composition“ ein.

A.5.22.11 footspace

Das Attribut **footspace** stellt den „footspace“ der „Page layout composition“ ein.

A.5.22.12 headheight

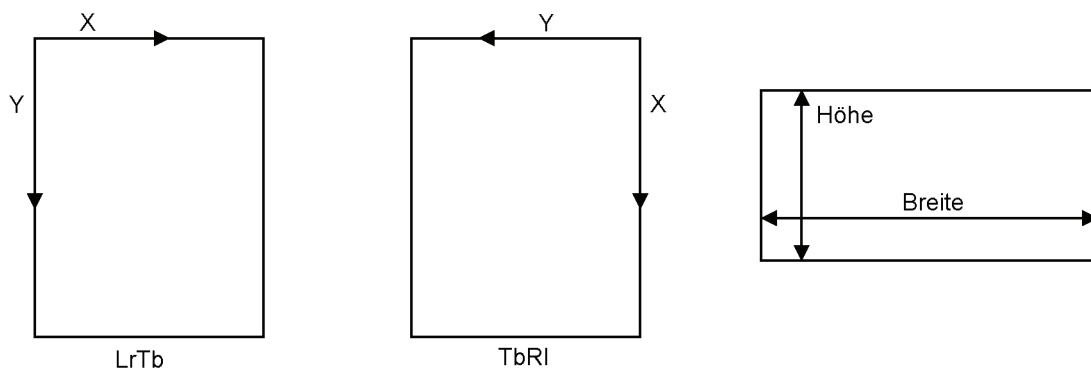
Das Attribut **headheight** stellt die „headheight“ der „Page layout composition“ ein.

A.5.22.13 headsep

Das Attribut **headsep** stellt den „headsep“ der „Page layout composition“ ein.

A.5.22.14 layout

Das Attribut **layout** bestimmt das Koordinatensystem des Elementes **Page**. Wenn „LrTb“ angegeben wird, ist der Nullpunkt links oben, die X-Achse ist nach rechts und die Y-Achse ist nach unten gerichtet. Wenn „TbRI“ angegeben wird, ist der Nullpunkt rechts oben, die X-Achse ist nach unten und die Y-Achse ist nach links gerichtet.



Das Koordinatensystem ändert sich in Abhängigkeit von dem angegebenen Layout. Breite und Höhe bleiben jedoch unverändert.

Bild A.6 – Attribut layout

A.5.22.15 oddsidemargin

Das Attribut **oddsidemargin** stellt bei einem Zweibildschirm-Anzeigegerät den linken Rand der „Page layout composition“ ein: im horizontalen Layout auf der rechten Seite und im vertikalen Layout auf der linken Seite.

Der Wert des Attributes **oddsidemargin** sollte verwendet werden. Bei einem Einbildschirm-Anzeigegerät wird als Wert von „oddsidemargin“ „evensidemargin“ benutzt.

A.5.22.16 textheight

Das Attribut **textheight** stellt die „textheight“ der „Page layout composition“ ein.

A.5.22.17 textwidth

Das Attribut **textwidth** stellt die „textwidth“ der „Page layout composition“ ein.

A.5.22.18 topmargin

Das Attribut **topmargin** stellt die „topmargin“ der „Page layout composition“ ein.

A.5.22.19 objid

Das Attribut **objid** gibt für jedes Element im Inhalt eine eindeutige Zeichenkette an.

A.5.23 Wait

Das Element **Wait** beschreibt den Ablauf der Bedienung des Timers.

Tabelle A.11 – Attribut von Wait

Attributname	Vorgabewert	Datentyp	Bemerkungen
time (erforderlich)	(Ganzzahl mit Vorzeichen)	Dezimalfolge	

Das Attribut **time** beschreibt den Timer. Wenn die Zeit „0“ ist, bleibt der Timer in Warteposition, auch für eine Eingabeoperation. Wenn die Zeit ein positiver Wert ist, geschieht während der eingestellten Zeit ($\times 100$ ms) nichts. Wenn die Zeit ein negativer Wert ist, geschieht während einer kürzeren Zeit als dem Absolutwert der eingestellten Zeit ($\times 100$ ms) nichts oder die Eingabeoperation wird ausgeführt.

A.5.24 TextBlock

Das Element **TextBlock** enthält die Angaben zur Einstellung von „Text“ und „Block“, um den „Block“ auf dem Bildschirm darzustellen. Für die Attribute des Elementes **TextBlock** wird empfohlen, nur die sich von der Information der Elemente **TextStyle** und **BlockStyle** (durch „textstyle“ und „blockstyle“ beschrieben) unterscheidende Information anzugeben.

Tabelle A.12 – Attribut von TextBlock (1 von 3)

Attributname	Vorgabewert	Datentyp	Bemerkungen
align		Zeichenkette	Gibt „head“, „center“ oder „bottom“ an.
baselineskip	(Ganzzahl mit Vorzeichen)	Dezimalfolge	[pt]*10
bgcolor		COLORREF-Zeichenkette	
bgimagemode		Zeichenkette	Gibt „fix“, „tile“ oder „centering“ an. Außerdem ist es erforderlich, gleichzeitig das Attribut refbgimage anzugeben.
blockheight	(Ganzzahl ohne Vorzeichen)	Dezimalfolge	[Pixel]
blockrule		Zeichenkette	Gibt als Attribute blockrule des Elementes TextBlock „horz-fixed“, „horz-adjustable“, „vert-fixed“ oder „vert-adjustable“ an.
blockstyle (erforderlich)		Stylefolge	Es ist erforderlich, die angegebene Style-Zeichenfolge der „stylelabel“ des genannten Elementes BlockStyle anzupassen.

Tabelle A.12 – Attribut von TextBlock (2 von 3)

Attributname	Vorgabewert	Datentyp	Bemerkungen
blockwidth	(Ganzzahl ohne Vorzeichen)	Dezimalfolge	[Pixel]
charspace	(Ganzzahl mit Vorzeichen)	Dezimalfolge	[pt]*10
column		Dezimalfolge	Gibt eine Ganzzahl zwischen „1“ und „9“ an.
columnsep	(Ganzzahl mit Vorzeichen)	Dezimalfolge	[pt]*10
empdotscode		Hexadezimalfolge	Gibt den Zeichencode an, der als Betonungspunkt benutzt wird. Außerdem ist es nötig, gleichzeitig die Attribute empdotsfontname und refempdotsfont anzugeben.
empdotsposition		Zeichenkette	Gibt „before“ oder „after“ an.
emlinetype		Zeichenkette	Gibt „none“, „solid“, „dotted“, „double“ oder „dashed“ an.
emlineposition		Dezimalfolge	Gibt „before“ oder „after“ an.
fontescapement		Dezimalfolge	Gibt „0“ oder „2700“ an.
fontfacename		Zeichenkette	Gibt den Fontnamen an.
fontorientation		Dezimalfolge	Gibt „0“ oder „2700“ an.
fontsize	(Ganzzahl mit Vorzeichen)	Dezimalfolge	[pt]*10
fontweight		Dezimalfolge	Gibt einen Wert zwischen „1“ und „1000“ an.
fontwidth	(Ganzzahl mit Vorzeichen)	Dezimalfolge	[pt]*10, gibt –10 an, wenn keine Änderung der Form der Schrift.
footskip	(vorzeichenlose Ganzzahl)	Dezimalfolge	[Pixel]
framecolor		COLORREF-Zeichenkette	
framemode		Zeichenkette	Gibt „curve“ oder „square“ an.
framewidth	(vorzeichenlose Ganzzahl)	Dezimalfolge	[Pixel]
layout		Zeichenkette	Gibt „LrTb“ oder „TbRI“ an.
letterspace	(Ganzzahl mit Vorzeichen)	Dezimalfolge	[pt]*10
linecolor		COLORREF-Zeichenkette	
linespace	(Ganzzahl mit Vorzeichen)	Dezimalfolge	[pt]*10
textlinewidth	(vorzeichenlose Ganzzahl)	Dezimalfolge	[pt]*10
objid (erforderlich)		Zeichenkette	Gibt die eindeutige Zeichenkette in der Datei an.

Tabelle A.12 – Attribut von TextBlock (3 von 3)

Attributname	Vorgabewert	Datentyp	Bemerkungen
parindent	(Ganzzahl mit Vorzeichen)	Dezimalfolge	[pt]*10
parskip	(Ganzzahl mit Vorzeichen)	Dezimalfolge	[pt]*10
refbgimage		Objekt-ID-Zeichenkette	Gibt die „objid“ des Elementes Image an. Außerdem ist es nötig, gleichzeitig das Attribut refimagemode anzugeben.
rubyalign		Zeichenkette	Gibt „start“ oder „center“ an.
rubyadjust		Zeichenkette	Gibt „line-edge“ oder „none“ an.
rubyoverhang		Zeichenkette	Gibt die Zeichenkette „auto“ oder „none“ an.
sidemargin	(vorzeichenlose Ganzzahl)	Dezimalfolge	[Pixel]
textbgcolor		COLORREF-Zeichenkette	
textcolor		COLORREF-Zeichenkette	
textstyle (erforderlich)		Style-Zeichenkette	Es ist erforderlich, die angegebene Style-Zeichenfolge dem „stylelabel“ des genannten Elementes TextStyle anzupassen.
toclabel			Gibt die Zeichenkette an, die im Inhaltsverzeichnis benutzt wird.
topskip	(vorzeichenlose Ganzzahl)	Dezimalfolge	[Pixel]
wordspace	(Ganzzahl mit Vorzeichen)	Dezimalfolge	[pt]*10
empdotsfontname		Zeichenkette	Gibt den Namen der für Betonungspunkte benutzten Schriftart an. Außerdem ist es nötig, gleichzeitig die Attribute empdotscode und refempdotsfont anzugeben.
refempdotsfont		Objekt-ID-Zeichenkette	Gibt die „objid“ des Elementes Font an. Außerdem ist es nötig, gleichzeitig die Attribute empdotscode und refempdotsname anzugeben.

A.5.24.1 fontsize

Das Attribut **fontsize** gibt die Schriftgröße an. Die Einheit ist 10 mal pt.

A.5.24.2 fontwidth

Das Attribut **fontwidth** gibt die Breite der Schriftart an. Üblicherweise wird die optimale Breite in Abhängigkeit der Schriftgröße ermittelt, dieses Attribut wird jedoch verwendet, wenn die Breite absichtlich geändert werden soll. Die Einheit ist 10 mal pt.

A.5.24.3 fontescapement

Das Attribut **fontescapement** gibt die Laufrichtung der Zeichen an: „0“ für eine horizontale Schriftart und „2700“ für eine vertikale Schriftart. Andere Spezifikationen werden nicht akzeptiert.

A.5.24.4 fontorientation

Das Attribut **fontorientation** gibt die Orientierung von gedrehten Zeichen an: „0“ für eine horizontale Schriftart und „2700“ für eine vertikale Schriftart. Andere Spezifikationen werden nicht akzeptiert.

A.5.24.5 fontfacename

Das Attribut **fontfacename** gibt den Namen der zu verwendenden Schriftart an. Wenn nichts angegeben ist, wird die Norm-Schriftart benutzt.

A.5.24.6 textcolor

Das Attribut **textcolor** gibt die Farbe des zu verwendenden Textes an. Wenn nichts angegeben ist, wird Schwarz (0x00000000) benutzt.

A.5.24.7 textbgcolor

Das Attribut **textbgcolor** gibt die Farbe des Hintergrundes des Textes an. Wenn nichts angegeben ist, wird normalerweise ein transparenter (0xff000000) Hintergrund verwendet.

A.5.24.8 wordspace

Das Attribut **wordspace** gibt an, wie breit der Abstand zwischen den Worten westlicher Sprachen ist. Die Einheit ist 10 mal pt.

A.5.24.9 letterspace

Das Attribut **letterspace** gibt die Größe des Abstandes zwischen den Buchstaben westlicher Sprachen an. Die Einheit ist 10 mal pt.

A.5.24.10 charspace

Das Attribut **charspace** gibt die Größe des Abstandes zwischen japanischen Zeichen an. Die Einheit ist 10 mal pt.

A.5.24.11 baselineskip

Das Attribut **baselineskip** gibt den Abstand zwischen den Grundlinien der Zeilen an. Die Einheit ist 10 mal pt.

A.5.24.12 linespace

Das Attribut **linespace** gibt den garantierten Mindestabstand zwischen den Zeilen (Durchschuss) an. Die Einheit ist 10 mal pt.

A.5.24.13 parindent

Das Attribut **parindent** gibt die Anfangsposition der ersten Zeile des Absatzes an. Ein negativer Wert bedeutet „overhanging“. Der maximale Wert für „overhanging“ ist gleich dem im Attribut **sidemargin** angegebenen Wert. Die Einheit ist 10 mal pt.

A.5.24.14 parskip

Das Attribut **parskip** gibt die Größe des Abstandes zwischen Absätzen an. Die Einheit ist 10 mal pt.

A.5.24.15 rubyalign

Das Attribut **rubyalign** beschreibt das Ausrichten des „Ruby Textes“. Die Werte sind nachstehend angegeben.

Tabelle A.13 – Attribut rubyalign

Wert für Ausrichten	
Auf Anfang ausgerichtet	Auf Mitte ausgerichtet
start	center

A.5.24.16 rubyadjust

Das Attribut **rubyadjust** beschreibt das Verfahren zur Anpassung des „Ruby Textes“ an den Zeilenrand. Die entsprechenden Werte sind nachstehend angegeben. Das Verfahren besteht darin, Anfang oder Ende des „Ruby Textes“ an den Zeilenanfang oder das Zeilenende anzupassen, wenn der „Ruby Text“ länger als die „Ruby Base“ ist.

Tabelle A.14 – Attribut rubyadjust

Wert für Zeilenumbruch	
Verfahren benutzt	Verfahren nicht benutzt
Zeilenrand	none

A.5.24.17 rubyoverhang

Das Attribut **rubyoverhang** gibt an, ob der Überstand des „Ruby Textes“ ausgeführt wird oder nicht, wenn der „Ruby Text“ länger als die „Ruby Base“ ist. Die entsprechenden Werte sind nachstehend angegeben.

Tabelle A.15 – Attribut rubyoverhang

Ruby-Überstand	Wert des Überstandes
Überstand	auto
Kein Überstand	none

A.5.24.18 empdotsposition

Das Attribut **empdotsposition** gibt die Position von Betonungspunkten an. Die Werte sind nachstehend angegeben.

Tabelle A.16 – Attribut empdotsposition

Betonungspunkt-Position	Positionswert
Davor (vertikales Layout: rechts, horizontales Layout: oben)	before
Danach (vertikales Layout: links, horizontales Layout: unten)	after

A.5.24.19 empdotscode

Das Attribut **empdotscode** gibt den Zeichen-(oder Symbol-)Code an, der als Betonungspunkt benutzt wird.

A.5.24.20 emplineposition

Das Attribut **emplineposition** gibt die Position der Betonungslinie an. Die Werte sind nachstehend angegeben.

Tabelle A.17 – Attribut emplineposition

Position der Betonungslinie	Positionswert
Davor (vertikales Layout: rechts, horizontales Layout: oben)	before
Danach (vertikales Layout: links, horizontales Layout: unten)	after

A.5.24.21 emlinetype

Das Attribut **emlinetype** gibt den Typ der Betonungslinie an. Die Werte sind nachstehend angegeben.

Tabelle A.18 – Attribut emlinetype

Linienart	Wert der Linienart
keine	none
durchgezogene Linie	solid
punktierte Linie	dotted
gestrichelte Linie	dashed
doppelte Linie	double

A.5.24.22 column

Das Attribut **column** gibt die Stufenzahl der Spalten an.

A.5.24.23 columnsep

Das Attribut **columnsep** gibt die Abstände zwischen den Spalten an. Die Einheit ist 10 mal pt.

A.5.24.24 align

Das Attribut **align** definiert die Ausrichtung der Textzeile.

Tabelle A.19 – Attribut align

Ausrichtung	Wert
am Kopf	head
in der Mitte	center
am Fuß	foot

A.5.24.25 textlinewidth

Das Attribut **textlinewidth** gibt die Linienbreite der Elemente **Empline** und **Box** an. Die Einheit ist 10 mal pt.

A.5.24.26 linecolor

Das Attribut **linecolor** gibt die Liniefarbe der Elemente **Empline** und **Box** an.

A.5.24.27 textstyle

Das Attribut **textstyle** gibt das „stylelabel“ des Elementes **TextStyle** an, das für die Wiedergabe des Textes erforderliche Angaben liefert.

A.5.24.28 fontweight

Das Attribut **fontweight** gibt die Schriftstärke des Textes an. Der Wert „400“ ist übliche Schriftstärke und „800“ ist „Bold“.

A.5.24.29 objid

Siehe A.5.22.19.

A.5.24.30 bgcolor

Das Attribut **bgcolor** gibt die Farbe des Hintergrundes des „Blocks“ an.

A.5.24.31 bgimagemode

Das Attribut **bgimagemode** gibt an, wie das im Attribut **refbgimage** angegebene Bild angeordnet werden sollte.

Der Wert ist „fix“: Die Bildposition liegt am Nullpunkt des „Blocks“.

Der Wert ist „tile“: Das Bild füllt den „Block“ vom Nullpunkt an aus.

Der Wert ist „center“: Die Bildposition liegt in der Mitte des „Blocks“.

A.5.24.32 blockheight

Das Attribut **blockheight** gibt die Höhe des „Blocks“ an.

A.5.24.33 blockrule

Das Attribut **blockrule** gibt an, wie stark die Abmessungen des „Blocks“ vergrößert werden müssen, wenn der Benutzer die Zeichengröße am Sichtgerät vergrößert. Die Werte für das Element **TextBlock** werden als „horz-fixed“, „horz-adjustable“, „vert-fixed“ und „vert-adjustable“ angegeben.

Tabelle A.20 – Attribut blockrule

Wert des Maßstabs	Maßstab
horz-fixed	Blockbreite kann nicht geändert werden.
horz-adjustable	Blockbreite darf geändert werden.
vert-fixed	Blockhöhe kann nicht geändert werden.
vert-adjustable	Blockhöhe darf geändert werden.
block-fixed	Blockbreite und -höhe können nicht geändert werden.
block-adjustable	Blockbreite und -höhe dürfen proportional geändert werden.

A.5.24.34 blockstyle

Das Attribut **blockstyle** gibt das „stylelabel“ des Elementes **BlockStyle** an, das die für die Wiedergabe des Textes erforderlichen Informationen enthält.

A.5.25.35 blockwidth

Das Attribut **blockwidth** gibt die Breite des „Blocks“ an.

A.5.24.36 footskip

Das Attribut **footskip** gibt den „footskip“ der „Block layout composition“ an.

A.5.24.37 framecolor

Das Attribut **framecolor** gibt die Rahmenfarbe des „Blocks“ an.

A.5.24.38 framemode

Das Attribut **framemode** gibt den Rahmentyp des „Blocks“ an. Es gibt zwei Möglichkeiten: „square“ bedeutet rechte Winkel an den vier Ecken und „curve“ bedeutet abgerundete Ecken.

A.5.24.39 framewidth

Das Attribut **framewidth** gibt die Rahmen-Linienbreite des „Blocks“ an. Wenn der Wert „0“ ist oder nicht angegeben, wird kein Rahmen wiedergegeben.

A.5.24.40 layout

Das Attribut **layout** bestimmt das Koordinatensystem des Elementes **TextBlock**. Wenn „LrTb“ angegeben wird, ist der Nullpunkt links oben, die X-Achse ist nach rechts und die Y-Achse ist nach unten gerichtet. Wenn „TbRl“ angegeben wird, ist der Nullpunkt rechts oben, die X-Achse ist nach unten und die Y-Achse ist nach links gerichtet.

A.5.24.41 refbgimage

Das Attribut **refbgimage** gibt die „objid“ des Elementes **Image** an, das als Hintergrund des Elementes **TextBlock** gezeigt wird. Wenn die „objid“ nicht angegeben ist, wird das Bild nicht angezeigt.

A.5.24.42 sidemargin

Das Attribut **sidemargin** stellt den Seitenrand der „Block layout composition“ ein.

A.5.24.43 toclabel

Das Attribut **toclabel** gibt die Zeichenkette an, die als Inhaltsverzeichnis auf dem Sichtgerät angezeigt wird. Wenn das „toclabel“ angegeben wird, nimmt das „Authoring tool“ die Information dieses Labels auf und setzt sie in die Information des Elementes **TOC** ein.

A.5.24.44 topskip

Das Attribut **topskip** gibt den „topskip“ der „Block layout composition“ an.

A.5.24.45 empdotsfontname

Das Attribut **empdotsfontname** gibt den Namen der Schriftart der Zeichen (oder Symbole) an, die als Betonungspunkte benutzt werden.

A.5.24.46 refempdotsfont

Das Attribut **refempdotsfont** gibt die „objid“ des Elementes **Font** an, das als Betonungspunkte benutzt wird. Wenn der Wert „0“ ist, wird die Schriftart des Haupttextes benutzt.

A.5.25 P

Das Element **P** gibt die kleinste Einheit innerhalb des Schriftstückes an. In einem allgemeinen Roman bedeutet dies „Paragraf“. Das Element **P** wird als kleinste Einheit behandelt, die elektronisch ausgedrückt werden kann.

Tabelle A.21 – Attribut von P

Attributname	Vorgabewert	Datentyp	Bemerkungen
refesound	0	Dezimalfolge	Gibt die „objid“ des Elementes eSound an.

Das Attribut **refesound** gibt die „objid“ des Elementes **eSound** an, das als eingebettete Tonwiedergabe benutzt wird. Wenn hier eingebetteter Ton angegeben wird, wird der Ton zur gleichen Zeit wiedergegeben, wie das Element **P** angezeigt wird.

A.5.26 Plot

Das Element **Plot** gibt das Inline-Bild und die Inline-Schaltfläche an.

Tabelle A.22 – Attribut von Plot

Attributname	Vorgabewert	Datentyp	Bemerkungen
adjustment	„bottom“	Zeichenkette	Gibt „center“, „baseline“, „top“ oder „bottom“ an.
refobj (erforderlich)		Objekt-ID-Zeichenkette	Gibt die „objid“ der Elemente Image oder Button an.
xsize (erforderlich)	vorzeichenlose Ganzzahl	Dezimalfolge	[pt]*10
ysize (erforderlich)	vorzeichenlose Ganzzahl	Dezimalfolge	[pt]*10

A.5.26.1 xsize

Das Attribut **xsize** gibt die Größe in Richtung der X-Achse des Inline-Bildes oder der Inline-Schaltfläche an. Die Einheit ist 10 mal pt.

A.5.26.2 ysize

Das Attribut **ysize** gibt die Größe in Richtung der Y-Achse des Inline-Bildes oder der Inline-Schaltfläche an. Die Einheit ist 10 mal pt.

A.5.26.3 refobj

Das Attribut **refobj** gibt die „objid“ der Elemente (**Image** und **Button**) an, angegeben als „inline“-Ziel.

A.5.26.4 adjustment

Das Attribut **adjustment** gibt die Position des Inline-Bildes oder der Inline-Schaltfläche für die Textzeile an.

Bei horizontalem Layout: Mit dem Wert „top“ werden die Oberkante des Inline-Bildes und der Inline-Schaltfläche sowie die Textzeile aufeinander abgestimmt. Der Wert „center“ dient zur Abstimmung der Mittellinien. Der Wert „basislinie“ bedeutet die Abstimmung der Grundlinien. Der Wert „bottom“ bedeutet Abstimmung der Unterkanten.

Bei vertikalem Layout: Mit dem Wert „top“ werden die rechte Kante des Inline-Bildes und der Inline-Schaltfläche sowie die Textzeile aufeinander abgestimmt. Der Wert „center“ dient zur Abstimmung der Mittellinien. Der Wert „basislinie“ bedeutet die Abstimmung der Grundlinien. Der Wert „bottom“ bedeutet Abstimmung der linken Kanten.

A.5.27 CR

Das Element **CR** gibt einen obligatorischen Zeilenumbruch an. Wenn dies unmittelbar nach dem Element **P** angegeben wird, ist der Grundlinienabstand die Summe von „baselineskip“ und „parskip“.

A.5.28 Fill

Das Element **Fill** gibt das längenvariable Zeichen für Durchschuss (Lead) an.

Tabelle A.23 – Attribut von Fill

Attributname	Vorgabewert	Datentyp	Bemerkungen
code (erforderlich)		Hexadezimalfolge	Kann zwischen „0x0020“ und „0xffff“ angegeben werden.

Das Attribut **code** gibt die Codenummer (UTF-16) des Zeichens an, das in dem Element **Fill** benutzt wird.

A.5.29 Space

Tabelle A.24 – Attribut von Space

Das Element **Space** beschreibt Abstand und Unterschneidung zwischen Zeichen.

Attributname	Vorgabewert	Datentyp	Bemerkungen
xsize (erforderlich)	(Ganzzahl mit Vorzeichen)	Dezimalfolge	[Pixel]

Das Attribut **xsize** gibt den Abstand in Richtung der X-Achse an. Ein negativer Wert bedeutet Unterschneidung.

A.5.30 CharButton

Das Element **CharButton** gibt den Bereich der Zeichenkette für die Funktion der Schaltfläche an.

Tabelle A.25 – Attribut von CharButton

Attributname	Vorgabewert	Datentyp	Bemerkungen
refobj (erforderlich)		Objekt-ID-Zeichenkette	Gibt die „objid“ des Elementes Button an.

Das Attribut **refobj** gibt die „objid“ des Elementes (Element **Button**) an, das als Ziel des „Button character“ angegeben ist.

A.5.31 Rubi

Das Element **Rubi** legt den „Ruby Text“ und die „Ruby Base“ fest.

A.5.32 Oyamoji

Das Element **Oyamoji** legt die „Ruby Base“ fest.

A.5.33 Gaiji

Das Element **Gaiji** gibt den Zeichencode (UTF-16) des externen Zeichens an und die alternative Zeichenfolge, wenn das externe Zeichen nicht wiedergegeben werden kann.

Tabelle A.26 – Attribut von Gaiji

Attributname	Vorgabewert	Datentyp	Bemerkungen
code (erforderlich)		Hexadezimalfolge	Kann zwischen „0x0020“ und „0xffff“ festgelegt werden.
fontfacename (erforderlich)		Zeichenkette	Gibt den Namen der Schriftart an.
refobj (erforderlich)		Objekt-ID-Zeichenkette	Gibt die „objid“ des Elementes Font an.

A.5.33.1 refobj

Das Attribut **refobj** gibt die „objid“ des Elementes (**Font**) an, das als Ziel des „Gaiji character“ angegeben wird.

A.5.33.2 fontfacename

Siehe A.5.24.5.

A.5.33.3 code

Das Attribut **code** gibt den Zeichencode an, der für externe Zeichen benutzt wird.

A.5.34 Altstring

Das Element **Altstring** gibt die alternative Zeichenkette an. Die alternative Zeichenkette wird benutzt, wenn die Anzeigeeinrichtung die Original-Zeichenkette nicht wiedergeben kann.

A.5.35 Org

Das Element **Org** gibt die Original-Zeichenkette an, die der Benutzer wünscht.

A.5.36 Alt

Das Element **Alt** gibt anstelle des Elementes **Org** die alternative Zeichenkette an.

A.5.37 Rubimoji

Das Element **Rubimoji** gibt den „Ruby Text“ für die „Ruby Base“ an, die in dem Element **Oyamoji** angegeben wird.

A.5.38 Box

Das Element **Box** gibt die Zeichenkette zur Umrandung mit einer rechteckförmigen Linie an.

Tabelle A.27 – Attribut von Box

Attributname	Vorgabewert	Datentyp	Bemerkungen
linetype (erforderlich)		Zeichenkette	„solid“, „dotted“, „dashed“ oder „double“ ist ausgewählt.

Das Attribut **linetype** beschreibt die Art der Linie des Rechtecks. Die Werte sind nachstehend angegeben.

Tabelle A.28 – Attribut linetype

Art der Linie	Wert Linien-Modus
durchgezogene Linie	solid
gepunktete Linie	dotted
gestrichelte Linie	dashed
doppelte Linie	double

A.5.39 EmpDots

Das Element **EmpDots** gibt die Zeichenkette an, die die Betonungspunkte hinzufügt.

Tabelle A.29 – Attribut von EmpDots

Attributname	Vorgabewert	Datentyp	Bemerkungen
empdotscode		Hexadezimalfolge	Gibt den Zeichencode an, der für einen Betonungspunkt verwendet wird. Außerdem ist es erforderlich, gleichzeitig die Attribute empdotsfontname und refempdotsfont anzugeben.
empdotsposition		Zeichenkette	Gibt „before“ oder „after“ an.
empdotsfontname		Zeichenkette	Gibt den Namen der Schriftart an, die für Betonungspunkte verwendet wird. Außerdem ist es erforderlich, gleichzeitig die Attribute empdotscode und empdotsname anzugeben.
refempdotsfont		Objekt-ID-Zeichenkette	Gibt die „objid“ des Elementes Font an. Außerdem ist es erforderlich, gleichzeitig die Attribute empdotscode und empdotsname anzugeben.

A.5.39.1 empdotsposition

Siehe A.5.24.18.

A.5.39.2 empdotscode

Siehe A.5.24.19.

A.5.39.3 empdotsfontname

Siehe A.5.24.45.

A.5.39.4 refempdotsfont

Siehe A.5.24.46.

A.5.40 EmpLine

Das Element **EmpLine** gibt die Zeichenkette an, durch die die Betonungslinie hinzugefügt wird.

Tabelle A.30 – Attribut von EmpLine

Attributname	Vorgabewert	Datentyp	Bemerkungen
emlinetype		Zeichenkette	Gibt „none“, „solid“, „dotted“, „dashed“ oder „double“ an.
emlineposition		Zeichenkette	Gibt „before“ oder „after“ an.

A.5.40.1 emlineposition

Siehe A.5.24.20.

A.5.40.2 emlinetype

Siehe A.5.24.21.

A.5.41 Sub

Das Element **Sub** gibt eine tiefgestellte Zeichenkette an.

A.5.42 Sup

Das Element **Sup** gibt eine hochgestellte Zeichenkette an.

A.5.43 Yoko

Das Element **Yoko** gibt die Zeichenkette an, deren Grundlinienrichtung im vertikalen Layout auf horizontal geändert wird.

A.5.44 Tate

Das Element **Tate** gibt die Zeichenkette westlicher Zeichen an, die im vertikalen Layout entgegen dem Uhrzeigersinn um 90° geschwenkt wird.

A.5.45 Nekase

Das Element **Nekase** gibt die Zeichenkette vertikaler Zeichen an, die im vertikalen Layout im Uhrzeigersinn um 90° geschwenkt wird.

A.5.46 NoBR

Das Element **NoBR** gibt die Zeichenkette an, die nicht geteilt werden darf.

A.5.47 DrawChar

Das Element **DrawChar** gibt das Zeichen als Initiale an.

Tabelle A.31 – Attribut von DrawChar

Attributname	Vorgabewert	Datentyp	Bemerkungen
line (erforderlich)	vorzeichenlose Ganzzahl	Dezimalfolge	Gibt die Anzahl der Zeilen an.

Das Attribut **line** gibt die Zeilenzahl für das Zeichen an, das als Initiale angegeben wird.

A.5.48 **Italic**

Das Element **Italic** legt die Zeichenfolge in kursiv fest.

A.5.49 **Bold**

Das Element **Bold** legt die Zeichenkette in fett fest.

A.5.50 **ImageBlock**

Das Element **ImageBlock** gibt die Information zur Einrichtung von „Block“ und „Image“, um den „Block“ auf dem Bildschirm darzustellen. Für die Attribute des Elementes **ImageBlock** wird empfohlen, nur die sich von der Information des Elementes **BlockStyle** (durch „blockstyle“ beschrieben) unterscheidende Information anzugeben. Wenn die Bilddatei nicht geöffnet werden kann, wird die alternative Zeichenkette angegeben.

Tabelle A.32 – Attribut von ImageBlock (1 von 2)

Attributname	Vorgabewert	Datentyp	Bemerkungen
x0 (erforderlich)	vorzeichenlose Ganzzahl	Dezimalfolge	[Pixel]
y0 (erforderlich)	vorzeichenlose Ganzzahl	Dezimalfolge	[Pixel]
x1 (erforderlich)	vorzeichenlose Ganzzahl	Dezimalfolge	[Pixel]
y1 (erforderlich)	vorzeichenlose Ganzzahl	Dezimalfolge	[Pixel]
xsize (erforderlich)	vorzeichenlose Ganzzahl	Dezimalfolge	[Pixel]
ysize (erforderlich)	vorzeichenlose Ganzzahl	Dezimalfolge	[Pixel]
refstream		Objekt-ID-Zeichenkette	Gibt die „objid“ des Elementes ImageStream. In dem Element Image kann nur „refstream“ gewählt werden.
objid (erforderlich)		Zeichenkette	Gibt in der Datei die eindeutige Zeichenfolge an.
bgcolor		COLORREF-Zeichenkette	
bgimagemode		Zeichenkette	Gibt „fix“, „tile“ oder „centering“ an. Außerdem ist es erforderlich, gleichzeitig das Attribut refbgimage anzugeben.
blockheight	vorzeichenlose Ganzzahl	Dezimalfolge	[Pixel]
blockrule		Zeichenkette	Für das ImageBlock-Attribut blockrule wird „blockfixed“ oder „block-adjustable“ angegeben.
blockstyle (erforderlich)		Style-Zeichenkette	Es ist erforderlich, die angegebene Style-Zeichenkette dem „stylelabel“ des genannten Elementes BlockStyle anzupassen.

Tabelle A.32 – Attribut von ImageBlock (2 von 2)

Attributname	Vorgabewert	Datentyp	Bemerkungen
blockwidth	vorzeichenlose Ganzzahl	Dezimalfolge	[Pixel]
foodskip	vorzeichenlose Ganzzahl	Dezimalfolge	[Pixel]
framecolor		COLORREF-Zeichenkette	
framemode		Zeichenkette	Gibt „curve“ oder „square“ an.
framewidth	vorzeichenlose Ganzzahl	Dezimalfolge	[Pixel]
layout		Zeichenkette	Gibt „LrTb“ oder „TbRl“ an.
refbgimage		Objekt-ID-Zeichenkette	Gibt die „objid“ des Elementes Image an. Die eigene ID kann jedoch nicht angegeben werden.
sidemargin	vorzeichenlose Ganzzahl	Dezimalfolge	[Pixel]
topskip	vorzeichenlose Ganzzahl	Dezimalfolge	[Pixel]

A.5.50.1 x0

Das Attribut **x0** bestimmt die X-Koordinate der oberen linken Ecke, die für das Auswählen des anzuzeigenden Teils des ursprünglichen Bildes benötigt wird.

A.5.50.2 y0

Das Attribut **y0** bestimmt die Y-Koordinate der oberen linken Ecke, die für das Auswählen des anzuzeigenden Teils des ursprünglichen Bildes benötigt wird.

A.5.50.3 x1

Das Attribut **x1** bestimmt die X-Koordinate der unteren rechten Ecke, die für das Auswählen des anzuzeigenden Teils des ursprünglichen Bildes benötigt wird.

A.5.50.4 y1

Das Attribut **y1** bestimmt die Y-Koordinate der unteren rechten Ecke, die für das Auswählen des anzuzeigenden Teils des ursprünglichen Bildes benötigt wird.

A.5.50.5 xsize

Das Attribut **xsize** legt die Größe des Bildausschnittes in X-Richtung fest. Dies bezieht sich auf die Vergrößerung entlang der X-Achse, wenn die Größe größer ist als (x1-x0), und auf die Verkleinerung entlang der X-Achse, wenn die X-Achse kürzer ist als (x1-x0).

A.5.50.6 ysize

Das Attribut **ysize** legt die Größe des Bildausschnittes in Y-Richtung fest. Dies bezieht sich auf die Vergrößerung entlang der Y-Achse, wenn die Größe größer ist als (y1-y0), und auf die Verkleinerung entlang der Y-Achse, wenn die Y-Achse kürzer ist als (y1-y0).

A.5.50.7 refstream

Das Attribut **refstream** gibt die „objid“ des Elementes **ImageStream** an, das die Daten des anzuzeigenden ursprünglichen Bildes enthält.

A.5.50.8 objid

Siehe A.5.22.19.

A.5.50.9 bgcolor

Siehe A.5.24.30.

A.5.50.10 bgimagemode

Siehe A.5.24.31.

A.5.50.11 blockheight

Siehe A.5.24.32.

A.5.50.12 blockrule

Das Attribut **blockrule** gibt an, wie die Block-Abmessungen vergrößert werden müssen, wenn der Benutzer auf dem Sichtgerät die Zeichengröße vergrößert. Die Werte für das Element **ImageBlock** werden als „block-fixed“ und „block-adjustable“ angegeben.

Bezüglich der Wertetabelle siehe A.5.24.33.

A.5.50.13 blockstyle

Siehe A.5.24.34.

A.5.50.14 blockwidth

Siehe A.5.24.35.

A.5.50.15 footskip

Siehe A.5.24.36.

A.5.50.16 framecolor

Siehe A.5.24.37.

A.5.50.17 framemode

Siehe A.5.24.38.

A.5.50.18 framewidth

Siehe A.5.24.39.

A.5.50.19 layout

Das Attribut **layout** gibt das Koordinatensystem des Elementes **ImageBlock** an. Wenn „LrTb“ angegeben wird, liegt der Nullpunkt links oben, die X-Achse ist nach rechts und die Y-Achse ist nach unten gerichtet. Wenn „TbRI“ angegeben wird, liegt der Nullpunkt rechts oben, die X-Achse ist nach unten und die Y-Achse ist nach links gerichtet.

A.5.50.20 refbgimage

Das Attribut **refbgimage** gibt die „objid“ des Elementes **Image** an, das als Hintergrund des Elementes **ImageBlock** gezeigt wird. Wenn die „objid“ nicht angegeben ist, wird das Bild nicht angezeigt.

A.5.50.21 sidemargin

Siehe A.5.24.42.

A.5.50.22 topskip

Siehe A.5.24.44.

A.5.51 ButtonBlock

Das Element **ButtonBlock** gibt die Information zum Einrichten von „Block“ und „Text“ an, um den „Block“ auf dem Bildschirm darzustellen. Für die Attribute des Elementes **ButtonBlock** wird empfohlen, nur die sich von der Information des Elementes **BlockStyle** (durch „blockstyle“ beschrieben) unterscheidende Information anzugeben. Auch das Element **ButtonBlock** beschreibt die Aktion zur Betätigung der Schaltfläche.

Es existieren 4 Zustände bei der Betätigung der Schaltfläche. Grundzustand (vor der Auswahl) → Focusin-Zustand (bei Auswahl) → Gedrückt-Zustand (wenn gedrückt) → Up-Zustand (nach dem Loslassen). Diese Zustände werden der Reihe nach durch eine Betätigung des Anwenders am Gerät geändert.

Tabelle A.33 – Attribut von ButtonBlock (1 von 2)

Attributname	Vorgabewert	Datentyp	Bemerkungen
objid (erforderlich)		Zeichenkette	Gibt die eindeutige Zeichenkette in der Datei an.
bgcolor		COLORREF-Zeichenkette	
bgimagemode		Zeichenkette	Gibt „fix“, „tile“ oder „centering“ an. Außerdem ist es erforderlich, gleichzeitig das Attribut refbgimage anzugeben.
blockheight	(Ganzzahl ohne Vorzeichen)	Dezimalfolge	[Pixel]
blockrule		Zeichenkette	Für das ButtonBlock-Attribut blockrule wird „blockfixed“ oder „block-adjustable“ angegeben.
blockstyle (erforderlich)		Style-Zeichenkette	Es ist erforderlich, die angegebene Style-Zeichenkette dem „stylelabel“ des genannten Elementes BlockStyle anzupassen.
blockwidth	(Ganzzahl ohne Vorzeichen)	Dezimalfolge	[Pixel]
foodskip	(Ganzzahl ohne Vorzeichen)	Dezimalfolge	[Pixel]
framecolor		COLORREF-Zeichenkette	
framemode		Zeichenkette	Gibt „curve“ oder „square“ an.
framewidth	(Ganzzahl ohne Vorzeichen)	Dezimalfolge	[Pixel]

Tabelle A.33 – Attribut von ButtonBlock (2 von 2)

Attributname	Vorgabewert	Datentyp	Bemerkungen
layout		Zeichenkette	Gibt „LrTb“ oder „TbRl“ an.
refbgimage		Objekt-ID-Zeichenkette	Gibt die „objid“ des Elementes Image an. Außerdem ist es erforderlich, gleichzeitig das Attribut bgimagemode anzugeben.
sidemargin	(Ganzzahl ohne Vorzeichen)	Dezimalfolge	[Pixel]
topskip	(Ganzzahl ohne Vorzeichen)	Dezimalfolge	[Pixel]

A.5.51.1 objid

Siehe A.5.22.19.

A.5.51.2 bgcolor

Siehe A.5.24.30.

A.5.51.3 bgimagemode

Siehe A.5.24.31.

A.5.51.4 blockheight

Siehe A.5.24.32.

A.5.51.5 blockrule

Das Attribut **blockrule** gibt an, wie die Block-Abmessungen vergrößert werden müssen, wenn der Benutzer auf dem Sichtgerät die Zeichengröße vergrößert. Die Werte für das Element **ButtonBlock** werden als „blockfixed“ und „block-adjustable“ angegeben.

Bezüglich der Wertetabelle siehe A.5.24.33.

A.5.51.6 blockstyle

Siehe A.5.24.34.

A.5.51.7 blockwidth

Siehe A.5.24.35.

A.5.51.8 footskip

Siehe A.5.24.36.

A.5.51.9 framecolor

Siehe A.5.24.37.

A.5.51.10 framemode

Siehe A.5.24.38.

A.5.51.11 framewidth

Siehe A.5.24.39.

A.5.51.12 layout

Das Attribut **layout** gibt das Koordinatensystem des Elementes **ButtonBlock** an. Wenn „LrTb“ angegeben wird, liegt der Nullpunkt links oben, die X-Achse ist nach rechts und die Y-Achse ist nach unten gerichtet. Wenn „TbRl“ angegeben wird, liegt der Nullpunkt rechts oben, die X-Achse ist nach unten und die Y-Achse ist nach links gerichtet.

A.5.51.13 refbgimage

Das Attribut **refbgimage** gibt die „objid“ des Elementes **Image** an, das als Hintergrund des Elementes **ButtonBlock** gezeigt wird. Wenn die „objid“ nicht angegeben ist, wird das Bild nicht angezeigt.

A.5.51.14 sidemargin

Siehe A.5.24.42.

A.5.51.15 topskip

Siehe A.5.24.44.

A.5.52 BaseButton

Das Element **BaseButton** beschreibt das Bild der Schaltfläche im Ruhezustand.

Tabelle A.34 – Attribut von BaseButton

Attributname	Vorgabewert	Datentyp	Bemerkungen
refimage		Objekt-ID-Zeichenkette	Gibt die Objekt-ID des Elementes Image an.

Das Element **refimage** gibt die Objekt-ID des als Schaltfläche angezeigten Elementes **Image** an. Wenn die Schaltfläche das Element CharButton ist, wird kein Bild angezeigt, auch wenn es angegeben ist.

A.5.53 FocusinButton

Das Element **FocusinButton** beschreibt Bild und Aktion der angewählten Schaltfläche. Es ist möglich, mehrere Schaltflächenbetätigungen festzulegen. Diese werden in der angegebenen Reihenfolge ausgeführt.

Tabelle A.35 – Attribut von FocusinButton

Attributname	Vorgabewert	Datentyp	Bemerkungen
refimage		Objekt-ID-Zeichenkette	Gibt die „objid“ des Elementes Image an.

Siehe A.5.52.