

Open Document Format for Office Applications (OpenDocument) v1.1

OASIS Standard, 1 Feb 2007

Specification URIs:

This Version:

<http://docs.oasis-open.org/office/v1.1/OS/OpenDocument-v1.1.odt>  
<http://docs.oasis-open.org/office/v1.1/OS/OpenDocument-v1.1.pdf>  
<http://docs.oasis-open.org/office/v1.1/OS/OpenDocument-v1.1.html.zip>

Previous Version:

<http://www.oasis-open.org/committees/download.php/19275/OpenDocument-v1.0ed2-cs1.odt>  
<http://www.oasis-open.org/committees/download.php/19274/OpenDocument-v1.0ed2-cs1.pdf>

Latest Version:

<http://docs.oasis-open.org/office/v1.1/OpenDocument-v1.1.odt>  
<http://docs.oasis-open.org/office/v1.1/OpenDocument-v1.1.pdf>  
<http://docs.oasis-open.org/office/v1.1/OpenDocument-v1.1.html.zip>

Latest Approved Version:

<http://docs.oasis-open.org/office/v1.1/OS/OpenDocument-v1.1.odt>  
<http://docs.oasis-open.org/office/v1.1/OS/OpenDocument-v1.1.pdf>  
<http://docs.oasis-open.org/office/v1.1/OS/OpenDocument-v1.1.html.zip>

Technical Committee:

OASIS Open Document Format for Office Applications (OpenDocument) TC

Chair:

Michael Brauer, Sun Microsystems, Inc.

Editors:

Patrick Durusau, Individual

Michael Brauer, Sun Microsystems, Inc.

Lars Oppermann, Sun Microsystems, Inc.

Related Work:

This specification supersedes OASIS OpenDocument v1.0.

Declared XML Namespaces:

A list of XML namespaces declared by this specification is available in section 1.3.

Abstract:

This is the specification of the Open Document Format for Office Applications (OpenDocument) format, an open, XML-based file format for office applications, based on OpenOffice.org XML [OOo].

Status:

This document was last revised or approved by the membership of OASIS on the above date. The level of approval is also listed above. Check the current location noted above for possible later revisions of this document. This document is updated periodically on no particular schedule.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at

[www.oasis-open.org/committees/office](http://www.oasis-open.org/committees/office).

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page

([www.oasis-open.org/committees/office/ipr.php](http://www.oasis-open.org/committees/office/ipr.php).

The non-normative errata page for this specification is located at [www.oasis-open.org/committees/office](http://www.oasis-open.org/committees/office).

Notices

Copyright © OASIS® 2002–2007. All Rights Reserved. OASIS trademark, IPR and other policies apply.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The names "OASIS", “OpenDocument”, “Open Document Format” and “ODF” are trademarks of [OASIS](http://www.oasis-open.org/), the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see http://www.oasis-open.org/who/trademark.php for above guidance.

Table of Contents

[1 Introduction 33](#1.Introduction|outline)

[1.1 Introduction 33](#1.1.Introduction|outline)

[1.2 Notation 33](#1.2.Notation|outline)

[1.3 Namespaces 33](#1.3.Namespaces|outline)

[1.4 Relax-NG Schema 35](#1.4.Relax-NG Schema|outline)

[1.5 Document Processing and Conformance 36](#1.5.Document Processing and Conformance|outline)

[1.6 White-Space Processing and EOL Handling 37](#1.6.White-Space Processing and EOL Handling|outline)

[1.7 MIME Types and File Name Extensions 37](#1.7.MIME Types and File Name Extensions|outline)

[2 Document Structure 39](#2.Document Structure|outline)

[2.1 Document Roots 39](#2.1.Document Roots|outline)

[2.1.1 Document Root Element Content Models 40](#2.1.1.Document Root Element Content Models|outline)

[2.1.2 Document Root Attributes 41](#2.1.2.Document Root Attributes|outline)

[2.2 Document Metadata 42](#2.2.Document Metadata|outline)

[2.2.1 Pre-Defined vs. Custom Metadata 42](#2.2.1.Pre-Defined vs. Custom Metadata|outline)

[2.2.2 Sample Metadata 43](#2.2.2.Sample Metadata|outline)

[2.3 Body Element and Document Types 43](#2.3.Body Element and Document Types|outline)

[2.3.1 Text Documents 44](#2.3.1.Text Documents|outline)

[2.3.2 Drawing Documents 46](#2.3.2.Drawing Documents|outline)

[2.3.3 Presentation Documents 47](#2.3.3.Presentation Documents|outline)

[2.3.4 Spreadsheet Documents 48](#2.3.4.Spreadsheet Documents|outline)

[2.3.5 Chart Documents 49](#2.3.5.Chart Documents|outline)

[2.3.6 Image Documents 49](#2.3.6.Image Documents|outline)

[2.4 Application Settings 50](#2.4.Application Settings|outline)

[2.4.1 Sequence of Settings 50](#2.4.1.Sequence of Settings|outline)

[2.4.2 Base Settings 51](#2.4.2.Base Settings|outline)

[2.4.3 Index Access of Sequences 52](#2.4.3.Index Access of Sequences|outline)

[2.4.4 Map Entry 52](#2.4.4.Map Entry|outline)

[2.4.5 Name Access of Sequences 53](#2.4.5.Name Access of Sequences|outline)

[2.4.6 Cursor Position Setting 53](#2.4.6.Cursor Position Setting|outline)

[2.5 Scripts 54](#2.5.Scripts|outline)

[2.5.1 Script 54](#2.5.1.Script|outline)

[2.6 Font Face Declarations 55](#2.6.Font Face Declarations|outline)

[2.7 Styles 55](#2.7.Styles|outline)

[2.7.1 Location of Styles 56](#2.7.1.Location of Styles |outline)

[2.8 Page Styles and Layout 58](#2.8.Page Styles and Layout|outline)

[3 Metadata Elements 59](#3.Metadata Elements|outline)

[3.1 Pre-Defined Metadata Elements 59](#3.1.Pre-Defined Metadata Elements|outline)

[3.1.1 Generator 59](#3.1.1.Generator|outline)

[3.1.2 Title 59](#3.1.2.Title|outline)

[3.1.3 Description 59](#3.1.3.Description|outline)

[3.1.4 Subject 60](#3.1.4.Subject|outline)

[3.1.5 Keywords 60](#3.1.5.Keywords|outline)

[3.1.6 Initial Creator 60](#3.1.6.Initial Creator |outline)

[3.1.7 Creator 60](#3.1.7.Creator|outline)

[3.1.8 Printed By 60](#3.1.8.Printed By|outline)

[3.1.9 Creation Date and Time 61](#3.1.9.Creation Date and Time|outline)

[3.1.10 Modification Date and Time 61](#3.1.10.Modification Date and Time|outline)

[3.1.11 Print Date and Time 61](#3.1.11.Print Date and Time|outline)

[3.1.12 Document Template 61](#3.1.12.Document Template|outline)

[3.1.13 Automatic Reload 62](#3.1.13.Automatic Reload|outline)

[3.1.14 Hyperlink Behavior 63](#3.1.14.Hyperlink Behavior|outline)

[3.1.15 Language 64](#3.1.15.Language|outline)

[3.1.16 Editing Cycles 64](#3.1.16.Editing Cycles|outline)

[3.1.17 Editing Duration 65](#3.1.17.Editing Duration|outline)

[3.1.18 Document Statistics 65](#3.1.18.Document Statistics|outline)

[3.2 User-defined Metadata 67](#3.2.User-defined Metadata|outline)

[3.3 Custom Metadata 68](#3.3.Custom Metadata|outline)

[4 Text Content 69](#4.Text Content|outline)

[4.1 Headings, Paragraphs and Basic Text Structure 69](#4.1.Headings, Paragraphs and Basic Text Structure|outline)

[4.1.1 Headings 69](#4.1.1.Headings|outline)

[4.1.2 Paragraphs 70](#4.1.2.Paragraphs|outline)

[4.1.3 Common Paragraph Elements Attributes 70](#4.1.3.Common Paragraph Elements Attributes|outline)

[4.2 Page Sequences 71](#4.2.Page Sequences|outline)

[4.2.1 Page 72](#4.2.1.Page|outline)

[4.3 Lists 73](#4.3.Lists|outline)

[4.3.1 List Block 73](#4.3.1.List Block|outline)

[4.3.2 List Item 74](#4.3.2.List Item|outline)

[4.3.3 List Header 76](#4.3.3.List Header|outline)

[4.3.4 Numbered Paragraphs 76](#4.3.4.Numbered Paragraphs|outline)

[4.4 Text Sections 77](#4.4.Text Sections|outline)

[4.4.1 Section Attributes 78](#4.4.1.Section Attributes|outline)

[4.4.2 Section Source 79](#4.4.2.Section Source|outline)

[4.4.3 DDE Source 80](#4.4.3.DDE Source|outline)

[4.5 Page-bound graphical content 81](#4.5.Page-bound graphical content|outline)

[4.6 Change Tracking 81](#4.6.Change Tracking|outline)

[4.6.1 Tracked Changes 81](#4.6.1.Tracked Changes|outline)

[4.6.2 Changed Regions 81](#4.6.2.Changed Regions|outline)

[4.6.3 Insertion 82](#4.6.3.Insertion|outline)

[4.6.4 Deletion 82](#4.6.4.Deletion|outline)

[4.6.5 Format Change 84](#4.6.5.Format Change|outline)

[4.6.6 Change Info 84](#4.6.6.Change Info|outline)

[4.6.7 Change Marks 85](#4.6.7.Change Marks|outline)

[4.7 Soft Page Break 85](#4.7.Soft Page Break|outline)

[4.8 Text Declarations 85](#4.8.Text Declarations|outline)

[5 Paragraph Elements Content 87](#5.Paragraph Elements Content|outline)

[5.1 Basic Text Content 87](#5.1.Basic Text Content|outline)

[5.1.1 White-space Characters 87](#5.1.1.White-space Characters|outline)

[5.1.2 Soft Hyphens, Hyphens, and Non-breaking Blanks 89](#5.1.2.Soft Hyphens, Hyphens, and Non-breaking Blanks|outline)

[5.1.3 Attributed Text 90](#5.1.3.Attributed Text|outline)

[5.1.4 Hyperlinks 90](#5.1.4.Hyperlinks|outline)

[5.2 Bookmarks and References 93](#5.2.Bookmarks and References|outline)

[5.2.1 Bookmarks 93](#5.2.1.Bookmarks|outline)

[5.2.2 References 93](#5.2.2.References|outline)

[5.3 Notes 95](#5.3.Notes|outline)

[5.3.1 Note Element 95](#5.3.1.Note Element|outline)

[5.4 Ruby 97](#5.4.Ruby|outline)

[5.5 Text Annotation 97](#5.5.Text Annotation|outline)

[5.6 Index Marks 98](#5.6.Index Marks|outline)

[5.7 Change Tracking and Change Marks 98](#5.7.Change Tracking and Change Marks|outline)

[5.8 Inline graphics and text-boxes 98](#5.8.Inline graphics and text-boxes|outline)

[6 Text Fields 99](#6.Text Fields|outline)

[6.1 Common Characteristics of Field Elements 99](#6.1.Common Characteristics of Field Elements|outline)

[6.2 Document Fields 100](#6.2.Document Fields|outline)

[6.2.1 Date Fields 100](#6.2.1.Date Fields|outline)

[6.2.2 Time Fields 101](#6.2.2.Time Fields|outline)

[6.2.3 Page Number Fields 103](#6.2.3.Page Number Fields|outline)

[6.2.4 Page Continuation Text 104](#6.2.4.Page Continuation Text|outline)

[6.2.5 Sender Fields 105](#6.2.5.Sender Fields|outline)

[6.2.6 Author Fields 108](#6.2.6.Author Fields|outline)

[6.2.7 Chapter Fields 108](#6.2.7.Chapter Fields|outline)

[6.2.8 File Name Fields 109](#6.2.8.File Name Fields|outline)

[6.2.9 Document Template Name Fields 110](#6.2.9.Document Template Name Fields|outline)

[6.2.10 Sheet Name Fields 111](#6.2.10.Sheet Name Fields|outline)

[6.3 Variable Fields 111](#6.3.Variable Fields|outline)

[6.3.1 Declaring Simple Variables 112](#6.3.1.Declaring Simple Variables|outline)

[6.3.2 Setting Simple Variables 112](#6.3.2.Setting Simple Variables|outline)

[6.3.3 Displaying Simple Variables 113](#6.3.3.Displaying Simple Variables|outline)

[6.3.4 Simple Variable Input Fields 114](#6.3.4.Simple Variable Input Fields|outline)

[6.3.5 Declaring User Variables 115](#6.3.5.Declaring User Variables|outline)

[6.3.6 Displaying User Variables 115](#6.3.6.Displaying User Variables|outline)

[6.3.7 User Variable Input Fields 116](#6.3.7.User Variable Input Fields|outline)

[6.3.8 Declaring Sequence Variables 117](#6.3.8.Declaring Sequence Variables|outline)

[6.3.9 Using Sequence Fields 118](#6.3.9.Using Sequence Fields|outline)

[6.3.10 Expression Fields 119](#6.3.10.Expression Fields|outline)

[6.3.11 Text Input Fields 120](#6.3.11.Text Input Fields|outline)

[6.4 Metadata Fields 120](#6.4.Metadata Fields|outline)

[6.4.1 Initial Creator 121](#6.4.1.Initial Creator|outline)

[6.4.2 Document Creation Date 121](#6.4.2.Document Creation Date|outline)

[6.4.3 Document Creation Time 121](#6.4.3.Document Creation Time|outline)

[6.4.4 Document Description 121](#6.4.4.Document Description|outline)

[6.4.5 User-Defined Document Information 122](#6.4.5.User-Defined Document Information|outline)

[6.4.6 Print Time 122](#6.4.6.Print Time|outline)

[6.4.7 Print Date 123](#6.4.7.Print Date|outline)

[6.4.8 Printed By 123](#6.4.8.Printed By|outline)

[6.4.9 Document Title 123](#6.4.9.Document Title|outline)

[6.4.10 Document Subject 123](#6.4.10.Document Subject|outline)

[6.4.11 Document Keywords 124](#6.4.11.Document Keywords|outline)

[6.4.12 Document Revision Number 124](#6.4.12.Document Revision Number|outline)

[6.4.13 Document Edit Duration 124](#6.4.13.Document Edit Duration|outline)

[6.4.14 Document Modification Time 124](#6.4.14.Document Modification Time|outline)

[6.4.15 Document Modification Date 125](#6.4.15.Document Modification Date|outline)

[6.4.16 Document Modified By 125](#6.4.16.Document Modified By|outline)

[6.4.17 Document Statistics Fields 125](#6.4.17.Document Statistics Fields|outline)

[6.5 Database Fields 126](#6.5.Database Fields|outline)

[6.5.1 Database Field Data Source 127](#6.5.1.Database Field Data Source|outline)

[6.5.2 Displaying Database Content 128](#6.5.2.Displaying Database Content|outline)

[6.5.3 Selecting the Next Database Row 128](#6.5.3.Selecting the Next Database Row|outline)

[6.5.4 Selecting a Row Number 129](#6.5.4.Selecting a Row Number|outline)

[6.5.5 Displaying the Row Number 130](#6.5.5.Displaying the Row Number|outline)

[6.5.6 Display Current Database and Table 131](#6.5.6.Display Current Database and Table|outline)

[6.6 More Fields 131](#6.6.More Fields|outline)

[6.6.1 Page Variable Fields 131](#6.6.1.Page Variable Fields|outline)

[6.6.2 Placeholders 132](#6.6.2.Placeholders|outline)

[6.6.3 Conditional Text Fields 133](#6.6.3.Conditional Text Fields|outline)

[6.6.4 Hidden Text Field 135](#6.6.4.Hidden Text Field|outline)

[6.6.5 Reference Fields 136](#6.6.5.Reference Fields|outline)

[6.6.6 Script Fields 138](#6.6.6.Script Fields|outline)

[6.6.7 Macro Fields 139](#6.6.7.Macro Fields|outline)

[6.6.8 Hidden Paragraph Fields 140](#6.6.8.Hidden Paragraph Fields|outline)

[6.6.9 DDE Connection Fields 141](#6.6.9.DDE Connection Fields|outline)

[6.6.10 Measure Fields 141](#6.6.10.Measure Fields|outline)

[6.6.11 Table Formula Field 142](#6.6.11.Table Formula Field|outline)

[6.7 Common Field Attributes 142](#6.7.Common Field Attributes|outline)

[6.7.1 Variable Value Types and Values 142](#6.7.1.Variable Value Types and Values|outline)

[6.7.2 Fixed 144](#6.7.2.Fixed|outline)

[6.7.3 Variable Name 145](#6.7.3.Variable Name|outline)

[6.7.4 Description 145](#6.7.4.Description |outline)

[6.7.5 Display 145](#6.7.5.Display|outline)

[6.7.6 Formula 146](#6.7.6.Formula|outline)

[6.7.7 Formatting Style 147](#6.7.7.Formatting Style|outline)

[6.7.8 Number Formatting Style 147](#6.7.8.Number Formatting Style|outline)

[7 Text Indices 149](#7.Text Indices|outline)

[7.1 Index Marks 149](#7.1.Index Marks|outline)

[7.1.1 Table of Content Index Marks 149](#7.1.1.Table of Content Index Marks|outline)

[7.1.2 User-Defined Index Marks 150](#7.1.2.User-Defined Index Marks|outline)

[7.1.3 Alphabetical Index Mark 151](#7.1.3.Alphabetical Index Mark|outline)

[7.1.4 Bibliography Index Mark 153](#7.1.4.Bibliography Index Mark|outline)

[7.2 Index Structure 154](#7.2.Index Structure|outline)

[7.2.1 Index Source 154](#7.2.1.Index Source|outline)

[7.2.2 Index Body Section 155](#7.2.2.Index Body Section|outline)

[7.2.3 Index Title Section 155](#7.2.3.Index Title Section|outline)

[7.3 Table Of Content 155](#7.3.Table Of Content|outline)

[7.3.1 Table of Content Source 156](#7.3.1.Table of Content Source|outline)

[7.3.2 Table of Content Entry Template 158](#7.3.2.Table of Content Entry Template|outline)

[7.4 Index of Illustrations 159](#7.4.Index of Illustrations|outline)

[7.4.1 Index of Illustration Source 160](#7.4.1.Index of Illustration Source|outline)

[7.4.2 Illustration Index Entry Template 161](#7.4.2.Illustration Index Entry Template|outline)

[7.5 Index of Tables 162](#7.5.Index of Tables|outline)

[7.5.1 Table Index Source 162](#7.5.1.Table Index Source|outline)

[7.5.2 Table Index Entry Template 163](#7.5.2.Table Index Entry Template|outline)

[7.6 Index of Objects 163](#7.6.Index of Objects|outline)

[7.6.1 Object Index Source 163](#7.6.1.Object Index Source|outline)

[7.6.2 Object Index Entry Template 165](#7.6.2.Object Index Entry Template|outline)

[7.7 User-Defined Index 165](#7.7.User-Defined Index|outline)

[7.7.1 User-Defined Index Source 165](#7.7.1.User-Defined Index Source|outline)

[7.7.2 User-Defined Index Entry Template 167](#7.7.2.User-Defined Index Entry Template|outline)

[7.8 Alphabetical Index 168](#7.8.Alphabetical Index|outline)

[7.8.1 Alphabetical Index Source 168](#7.8.1.Alphabetical Index Source|outline)

[7.8.2 Auto Mark File 172](#7.8.2.Auto Mark File|outline)

[7.8.3 Alphabetical Index Entry Template 172](#7.8.3.Alphabetical Index Entry Template|outline)

[7.9 Bibliography 173](#7.9.Bibliography|outline)

[7.9.1 Bibliography Index Source 173](#7.9.1.Bibliography Index Source|outline)

[7.9.2 Bibliography Entry Template 174](#7.9.2.Bibliography Entry Template|outline)

[7.10 index source styles 174](#7.10.index source styles|outline)

[7.10.1 Index source style 175](#7.10.1.Index source style|outline)

[7.11 Index title template 175](#7.11.Index title template|outline)

[7.12 Index Template Entries 175](#7.12.Index Template Entries|outline)

[7.12.1 Chapter Information 176](#7.12.1.Chapter Information|outline)

[7.12.2 Entry Text 176](#7.12.2.Entry Text|outline)

[7.12.3 Page Number 176](#7.12.3.Page Number|outline)

[7.12.4 Fixed String 177](#7.12.4.Fixed String|outline)

[7.12.5 Bibliography Information 177](#7.12.5.Bibliography Information|outline)

[7.12.6 Tab Stop 178](#7.12.6.Tab Stop|outline)

[7.12.7 Hyperlink Start and End 179](#7.12.7.Hyperlink Start and End|outline)

[7.12.8 Example of an Index Entry Configuration 180](#7.12.8.Example of an Index Entry Configuration|outline)

[8 Tables 182](#8.Tables|outline)

[8.1 Basic Table Model 182](#8.1.Basic Table Model|outline)

[8.1.1 Table Element 182](#8.1.1.Table Element|outline)

[8.1.2 Table Row 186](#8.1.2.Table Row |outline)

[8.1.3 Table Cell 188](#8.1.3.Table Cell |outline)

[8.2 Advanced Table Model 193](#8.2.Advanced Table Model|outline)

[8.2.1 Column Description 193](#8.2.1.Column Description |outline)

[8.2.2 Header Columns 195](#8.2.2.Header Columns|outline)

[8.2.3 Column Groups 195](#8.2.3.Column Groups|outline)

[8.2.4 Header Rows 196](#8.2.4.Header Rows|outline)

[8.2.5 Row Groups 196](#8.2.5.Row Groups|outline)

[8.2.6 Subtables 197](#8.2.6.Subtables |outline)

[8.3 Advanced Tables 200](#8.3.Advanced Tables|outline)

[8.3.1 Referencing Table Cells 200](#8.3.1.Referencing Table Cells|outline)

[8.3.2 Linked Tables 202](#8.3.2.Linked Tables|outline)

[8.3.3 Scenario Tables 204](#8.3.3.Scenario Tables|outline)

[8.3.4 Shapes 206](#8.3.4.Shapes|outline)

[8.4 Advanced Table Cells 207](#8.4.Advanced Table Cells |outline)

[8.4.1 Linked Table Cells 207](#8.4.1.Linked Table Cells|outline)

[8.4.2 Cell Annotation 208](#8.4.2.Cell Annotation |outline)

[8.4.3 Detective 208](#8.4.3.Detective|outline)

[8.4.4 Detective Operation 208](#8.4.4.Detective Operation|outline)

[8.4.5 Highlighted Range 209](#8.4.5.Highlighted Range|outline)

[8.5 Spreadsheet Document Content 211](#8.5.Spreadsheet Document Content|outline)

[8.5.1 Document Protection 211](#8.5.1.Document Protection|outline)

[8.5.2 Calculation Settings 211](#8.5.2.Calculation Settings|outline)

[8.5.3 Table Cell Content Validations 214](#8.5.3.Table Cell Content Validations|outline)

[8.5.4 Label Ranges 218](#8.5.4.Label Ranges|outline)

[8.5.5 Named Expressions 219](#8.5.5.Named Expressions|outline)

[8.6 Database Ranges 221](#8.6.Database Ranges|outline)

[8.6.1 Database Range 222](#8.6.1.Database Range |outline)

[8.6.2 Database Source SQL 225](#8.6.2.Database Source SQL |outline)

[8.6.3 Database Source Table 225](#8.6.3.Database Source Table |outline)

[8.6.4 Database Source Query 226](#8.6.4.Database Source Query|outline)

[8.6.5 Sort 227](#8.6.5.Sort |outline)

[8.6.6 Sort By 228](#8.6.6.Sort By |outline)

[8.6.7 Subtotal Rules 229](#8.6.7.Subtotal Rules |outline)

[8.6.8 Subtotal Sort Groups 230](#8.6.8.Subtotal Sort Groups |outline)

[8.6.9 Subtotal Rule 231](#8.6.9.Subtotal Rule |outline)

[8.6.10 Subtotal Field 232](#8.6.10.Subtotal Field |outline)

[8.7 Filters 233](#8.7.Filters|outline)

[8.7.1 Table Filter 233](#8.7.1.Table Filter|outline)

[8.7.2 Filter And 234](#8.7.2.Filter And|outline)

[8.7.3 Filter Or 235](#8.7.3.Filter Or |outline)

[8.7.4 Filter Condition 235](#8.7.4.Filter Condition |outline)

[8.8 Data Pilot Tables 237](#8.8.Data Pilot Tables|outline)

[8.8.1 Data Pilot Table 237](#8.8.1.Data Pilot Table|outline)

[8.8.2 Source Cell Range 242](#8.8.2.Source Cell Range|outline)

[8.8.3 Source Service 243](#8.8.3.Source Service|outline)

[8.8.4 Data Pilot Field 244](#8.8.4.Data Pilot Field|outline)

[8.8.5 Data Pilot Level 246](#8.8.5.Data Pilot Level|outline)

[8.8.6 Data Pilot Subtotals 247](#8.8.6.Data Pilot Subtotals|outline)

[8.8.7 Data Pilot Subtotal 247](#8.8.7.Data Pilot Subtotal|outline)

[8.8.8 Data Pilot Members 248](#8.8.8.Data Pilot Members|outline)

[8.8.9 Data Pilot Member 248](#8.8.9.Data Pilot Member|outline)

[8.8.10 Data Pilot Display Info 249](#8.8.10.Data Pilot Display Info|outline)

[8.8.11 Data Pilot Sort Info 250](#8.8.11.Data Pilot Sort Info|outline)

[8.8.12 Data Pilot Layout Info 251](#8.8.12.Data Pilot Layout Info|outline)

[8.8.13 Data Pilot Field Reference 252](#8.8.13.Data Pilot Field Reference|outline)

[8.8.14 Data Pilot Groups 254](#8.8.14.Data Pilot Groups|outline)

[8.8.15 Data Pilot Group 256](#8.8.15.Data Pilot Group|outline)

[8.8.16 Data Pilot Group Member 257](#8.8.16.Data Pilot Group Member|outline)

[8.9 Consolidation 257](#8.9.Consolidation|outline)

[8.10 DDE Links 259](#8.10.DDE Links|outline)

[8.11 Change Tracking in Spreadsheets 259](#8.11.Change Tracking in Spreadsheets|outline)

[8.11.1 Tracked Changes 259](#8.11.1.Tracked Changes|outline)

[8.11.2 Insertion 260](#8.11.2.Insertion|outline)

[8.11.3 Dependencies 261](#8.11.3.Dependencies|outline)

[8.11.4 Dependence 261](#8.11.4.Dependence|outline)

[8.11.5 Deletions 262](#8.11.5.Deletions|outline)

[8.11.6 Cell Content Deletion 262](#8.11.6.Cell Content Deletion|outline)

[8.11.7 Change Deletion 262](#8.11.7.Change Deletion|outline)

[8.11.8 Deletion 263](#8.11.8.Deletion|outline)

[8.11.9 Cut Offs 264](#8.11.9.Cut Offs|outline)

[8.11.10 Insertion Cut Off 265](#8.11.10.Insertion Cut Off|outline)

[8.11.11 Movement Cut Off 265](#8.11.11.Movement Cut Off|outline)

[8.11.12 Movement 266](#8.11.12.Movement|outline)

[8.11.13 Target Range Address, Source Range Address 267](#8.11.13.Target Range Address, Source Range Address|outline)

[8.11.14 Change Track Cell 268](#8.11.14.Change Track Cell|outline)

[8.11.15 Cell Content Change 269](#8.11.15.Cell Content Change|outline)

[8.11.16 Cell Address 270](#8.11.16.Cell Address|outline)

[8.11.17 Previous 270](#8.11.17.Previous|outline)

[8.11.18 Common Change Tracking Attributes 270](#8.11.18.Common Change Tracking Attributes|outline)

[9 Graphic Content 272](#9.Graphic Content|outline)

[9.1 Enhanced Page Features for Graphical Applications 272](#9.1.Enhanced Page Features for Graphical Applications|outline)

[9.1.1 Handout Master 272](#9.1.1.Handout Master|outline)

[9.1.2 Layer Sets 273](#9.1.2.Layer Sets|outline)

[9.1.3 Layer 274](#9.1.3.Layer|outline)

[9.1.4 Drawing Pages 275](#9.1.4.Drawing Pages|outline)

[9.1.5 Presentation Notes 278](#9.1.5.Presentation Notes|outline)

[9.2 Drawing Shapes 279](#9.2.Drawing Shapes|outline)

[9.2.1 Rectangle 279](#9.2.1.Rectangle|outline)

[9.2.2 Line 280](#9.2.2.Line|outline)

[9.2.3 Polyline 281](#9.2.3.Polyline|outline)

[9.2.4 Polygon 282](#9.2.4.Polygon|outline)

[9.2.5 Regular Polygon 283](#9.2.5.Regular Polygon|outline)

[9.2.6 Path 285](#9.2.6.Path|outline)

[9.2.7 Circle 286](#9.2.7.Circle|outline)

[9.2.8 Ellipse 288](#9.2.8.Ellipse|outline)

[9.2.9 Connector 289](#9.2.9.Connector|outline)

[9.2.10 Caption 293](#9.2.10.Caption|outline)

[9.2.11 Measure 294](#9.2.11.Measure|outline)

[9.2.12 Control 295](#9.2.12.Control|outline)

[9.2.13 Page Thumbnail 296](#9.2.13.Page Thumbnail|outline)

[9.2.14 Grouping 297](#9.2.14.Grouping|outline)

[9.2.15 Common Drawing Shape Attributes 298](#9.2.15.Common Drawing Shape Attributes|outline)

[9.2.16 Common Shape Attributes for Text and Spreadsheet Documents 302](#9.2.16.Common Shape Attributes for Text and Spreadsheet Documents|outline)

[9.2.17 Common Drawing Shape Content 305](#9.2.17.Common Drawing Shape Content|outline)

[9.2.18 Common Shape Attribute Groups 305](#9.2.18.Common Shape Attribute Groups|outline)

[9.2.19 Glue Points 305](#9.2.19.Glue Points|outline)

[9.2.20 Title and Description 307](#9.2.20.Title and Description|outline)

[9.2.21 Event Listeners 308](#9.2.21.Event Listeners|outline)

[9.3 Frames 308](#9.3.Frames|outline)

[9.3.1 Text Box 311](#9.3.1.Text Box|outline)

[9.3.2 Image 313](#9.3.2.Image|outline)

[9.3.3 Objects 314](#9.3.3.Objects|outline)

[9.3.4 Applet 316](#9.3.4.Applet|outline)

[9.3.5 Plugins 318](#9.3.5.Plugins|outline)

[9.3.6 Parameters 319](#9.3.6.Parameters|outline)

[9.3.7 Floating Frame 319](#9.3.7.Floating Frame|outline)

[9.3.8 Contour 320](#9.3.8.Contour|outline)

[9.3.9 Hyperlinks 321](#9.3.9.Hyperlinks|outline)

[9.3.10 Client Side Image Maps 323](#9.3.10.Client Side Image Maps|outline)

[9.4 3D Shapes 327](#9.4.3D Shapes|outline)

[9.4.1 Scene 327](#9.4.1.Scene|outline)

[9.4.2 Light 331](#9.4.2.Light|outline)

[9.4.3 Cube 332](#9.4.3.Cube|outline)

[9.4.4 Sphere 333](#9.4.4.Sphere|outline)

[9.4.5 Extrude 334](#9.4.5.Extrude|outline)

[9.4.6 Rotate 334](#9.4.6.Rotate|outline)

[9.5 Custom Shape 335](#9.5.Custom Shape|outline)

[9.5.1 Enhanced Geometry 336](#9.5.1.Enhanced Geometry|outline)

[9.5.2 Enhanced Geometry - Extrusion Attributes 339](#9.5.2.Enhanced Geometry - Extrusion Attributes|outline)

[9.5.3 Enhanced Geometry - Path Attributes 345](#9.5.3.Enhanced Geometry - Path Attributes|outline)

[9.5.4 Enhanced Geometry - Text Path Attributes 349](#9.5.4.Enhanced Geometry - Text Path Attributes|outline)

[9.5.5 Enhanced Geometry – Equation 350](#9.5.5.Enhanced Geometry – Equation|outline)

[9.5.6 Enhanced Geometry - Handle Attributes 352](#9.5.6.Enhanced Geometry - Handle Attributes|outline)

[9.6 Presentation Shapes 356](#9.6.Presentation Shapes|outline)

[9.6.1 Common Presentation Shape Attributes 356](#9.6.1.Common Presentation Shape Attributes|outline)

[9.7 Presentation Animations 358](#9.7.Presentation Animations|outline)

[9.7.1 Sound 359](#9.7.1.Sound|outline)

[9.7.2 Show Shape 360](#9.7.2.Show Shape|outline)

[9.7.3 Show Text 363](#9.7.3.Show Text|outline)

[9.7.4 Hide Shape 364](#9.7.4.Hide Shape|outline)

[9.7.5 Hide Text 364](#9.7.5.Hide Text|outline)

[9.7.6 Dim 364](#9.7.6.Dim|outline)

[9.7.7 Play 365](#9.7.7.Play|outline)

[9.7.8 Effect groups 365](#9.7.8.Effect groups|outline)

[9.8 SMIL Presentation Animations 365](#9.8.SMIL Presentation Animations|outline)

[9.8.1 Recommended Usage Of SMIL 366](#9.8.1.Recommended Usage Of SMIL|outline)

[9.8.2 Document Dependent SMIL Animation Attribute Values 367](#9.8.2.Document Dependent SMIL Animation Attribute Values|outline)

[9.8.3 SMIL Presentation Animation Attributes 369](#9.8.3.SMIL Presentation Animation Attributes|outline)

[9.9 Presentation Events 372](#9.9.Presentation Events|outline)

[9.10 Presentation Text Fields 375](#9.10.Presentation Text Fields|outline)

[9.10.1 Header Field 375](#9.10.1.Header Field|outline)

[9.10.2 Footer Field 375](#9.10.2.Footer Field|outline)

[9.10.3 Date and Time Field 375](#9.10.3.Date and Time Field|outline)

[9.11 Presentation Document Content 376](#9.11.Presentation Document Content|outline)

[9.11.1 Presentation Declarations 376](#9.11.1.Presentation Declarations|outline)

[9.11.2 Header field declaration 376](#9.11.2.Header field declaration|outline)

[9.11.3 Footer field declaration 376](#9.11.3.Footer field declaration|outline)

[9.11.4 Date and Time field declaration 377](#9.11.4.Date and Time field declaration|outline)

[9.11.5 Presentation Settings 377](#9.11.5.Presentation Settings|outline)

[9.11.6 Show Definitions 381](#9.11.6.Show Definitions|outline)

[10 Chart Content 383](#10.Chart Content|outline)

[10.1 Introduction to Chart Documents 383](#10.1.Introduction to Chart Documents|outline)

[10.2 Chart 383](#10.2.Chart|outline)

[10.3 Title, Subtitle and Footer 387](#10.3.Title, Subtitle and Footer|outline)

[10.3.1 Title 387](#10.3.1.Title|outline)

[10.3.2 Subtitle 388](#10.3.2.Subtitle|outline)

[10.3.3 Footer 388](#10.3.3.Footer|outline)

[10.4 Legend 388](#10.4.Legend|outline)

[10.5 Plot Area 390](#10.5.Plot Area|outline)

[10.5.1 3D Plot Area 392](#10.5.1.3D Plot Area|outline)

[10.6 Wall 392](#10.6.Wall|outline)

[10.7 Floor 393](#10.7.Floor|outline)

[10.8 Axis 394](#10.8.Axis|outline)

[10.8.1 Grid 395](#10.8.1.Grid|outline)

[10.9 Series 396](#10.9.Series|outline)

[10.9.1 Domain 398](#10.9.1.Domain|outline)

[10.10 Categories 398](#10.10.Categories|outline)

[10.11 Data Point 398](#10.11.Data Point|outline)

[10.12 Mean Value 399](#10.12.Mean Value|outline)

[10.13 Error Indicator 400](#10.13.Error Indicator|outline)

[10.14 Regression Curves 400](#10.14.Regression Curves|outline)

[10.14.1 Stock Chart Markers 401](#10.14.1.Stock Chart Markers|outline)

[11 Form Content 402](#11.Form Content|outline)

[11.1 Form 403](#11.1.Form|outline)

[11.1.1 Action 404](#11.1.1.Action|outline)

[11.1.2 Target Frame 404](#11.1.2.Target Frame|outline)

[11.1.3 Method 405](#11.1.3.Method|outline)

[11.1.4 Encoding Type 405](#11.1.4.Encoding Type|outline)

[11.1.5 Allow Deletes 405](#11.1.5.Allow Deletes|outline)

[11.1.6 Allow Inserts 406](#11.1.6.Allow Inserts|outline)

[11.1.7 Allow Updates 406](#11.1.7.Allow Updates|outline)

[11.1.8 Apply Filter 406](#11.1.8.Apply Filter|outline)

[11.1.9 Command Type 406](#11.1.9.Command Type|outline)

[11.1.10 Command 407](#11.1.10.Command|outline)

[11.1.11 Data Source 407](#11.1.11.Data Source|outline)

[11.1.12 Master Fields 407](#11.1.12.Master Fields|outline)

[11.1.13 Detail Fields 408](#11.1.13.Detail Fields|outline)

[11.1.14 Escape Processing 408](#11.1.14.Escape Processing|outline)

[11.1.15 Filter 408](#11.1.15.Filter|outline)

[11.1.16 Ignore Result 408](#11.1.16.Ignore Result|outline)

[11.1.17 Navigation Mode 409](#11.1.17.Navigation Mode|outline)

[11.1.18 Order 409](#11.1.18.Order|outline)

[11.1.19 Tabbing Cycle 409](#11.1.19.Tabbing Cycle|outline)

[11.1.20 Connection Resource 410](#11.1.20.Connection Resource|outline)

[11.2 XForms Model 410](#11.2.XForms Model|outline)

[11.2.1 XForms Model 411](#11.2.1.XForms Model|outline)

[11.3 Controls 411](#11.3.Controls|outline)

[11.3.1 Text 412](#11.3.1.Text|outline)

[11.3.2 Text Area 412](#11.3.2.Text Area|outline)

[11.3.3 Password 413](#11.3.3.Password|outline)

[11.3.4 File 414](#11.3.4.File|outline)

[11.3.5 Formatted Text 414](#11.3.5.Formatted Text|outline)

[11.3.6 Number 416](#11.3.6.Number|outline)

[11.3.7 Date And Time 417](#11.3.7.Date And Time|outline)

[11.3.8 Fixed Text 419](#11.3.8.Fixed Text|outline)

[11.3.9 Combo Box 419](#11.3.9.Combo Box|outline)

[11.3.10 List Box 421](#11.3.10.List Box|outline)

[11.3.11 Button 422](#11.3.11.Button|outline)

[11.3.12 Image 424](#11.3.12.Image|outline)

[11.3.13 Check Box 424](#11.3.13.Check Box|outline)

[11.3.14 Radio Button 426](#11.3.14.Radio Button|outline)

[11.3.15 Frame 427](#11.3.15.Frame|outline)

[11.3.16 Image Frame 427](#11.3.16.Image Frame|outline)

[11.3.17 Hidden 427](#11.3.17.Hidden|outline)

[11.3.18 Grid 428](#11.3.18.Grid|outline)

[11.3.19 Value Range 429](#11.3.19.Value Range|outline)

[11.3.20 Generic Control 431](#11.3.20.Generic Control|outline)

[11.4 Common Form and Control Attributes 431](#11.4.Common Form and Control Attributes|outline)

[11.4.1 Name 431](#11.4.1.Name|outline)

[11.4.2 Control Implementation 432](#11.4.2.Control Implementation|outline)

[11.4.3 Bind to XForms 432](#11.4.3.Bind to XForms|outline)

[11.5 Common Control Attributes 432](#11.5.Common Control Attributes|outline)

[11.5.1 Button Type 432](#11.5.1.Button Type|outline)

[11.5.2 Control ID 433](#11.5.2.Control ID|outline)

[11.5.3 Current Selected 434](#11.5.3.Current Selected|outline)

[11.5.4 Value and Current Value 434](#11.5.4.Value and Current Value|outline)

[11.5.5 Disabled 435](#11.5.5.Disabled|outline)

[11.5.6 Dropdown 436](#11.5.6.Dropdown|outline)

[11.5.7 For 436](#11.5.7.For|outline)

[11.5.8 Image Data 437](#11.5.8.Image Data|outline)

[11.5.9 Label 437](#11.5.9.Label|outline)

[11.5.10 Maximum Length 437](#11.5.10.Maximum Length|outline)

[11.5.11 Printable 438](#11.5.11.Printable|outline)

[11.5.12 Read only 439](#11.5.12.Read only|outline)

[11.5.13 Selected 439](#11.5.13.Selected|outline)

[11.5.14 Size 439](#11.5.14.Size|outline)

[11.5.15 Tab Index 440](#11.5.15.Tab Index|outline)

[11.5.16 Tab Stop 441](#11.5.16.Tab Stop|outline)

[11.5.17 Target Frame 441](#11.5.17.Target Frame|outline)

[11.5.18 Target Location 442](#11.5.18.Target Location|outline)

[11.5.19 Title 442](#11.5.19.Title|outline)

[11.5.20 Visual Effect 443](#11.5.20.Visual Effect|outline)

[11.5.21 Relative Image Position 443](#11.5.21.Relative Image Position|outline)

[11.5.22 Database Binding Attributes 444](#11.5.22.Database Binding Attributes|outline)

[11.6 Event Listeners 446](#11.6.Event Listeners|outline)

[11.7 Properties 448](#11.7.Properties|outline)

[11.7.1 Property Set 448](#11.7.1.Property Set|outline)

[11.7.2 Property 448](#11.7.2.Property|outline)

[11.7.3 List Property 449](#11.7.3.List Property|outline)

[12 Common Content 452](#12.Common Content|outline)

[12.1 Annotation 452](#12.1.Annotation|outline)

[12.1.1 Creator 453](#12.1.1.Creator|outline)

[12.1.2 Creation Date and Time 453](#12.1.2.Creation Date and Time|outline)

[12.1.3 Creation Date and Time String 453](#12.1.3.Creation Date and Time String|outline)

[12.2 Number Format 453](#12.2.Number Format|outline)

[12.2.1 Prefix and Suffix 453](#12.2.1.Prefix and Suffix|outline)

[12.2.2 Format Specification 454](#12.2.2.Format Specification|outline)

[12.2.3 Letter Synchronization in Number Formats 454](#12.2.3.Letter Synchronization in Number Formats|outline)

[12.3 Change Tracking Metadata 455](#12.3.Change Tracking Metadata|outline)

[12.4 Event Listener Tables 455](#12.4.Event Listener Tables|outline)

[12.4.1 Event Listener 456](#12.4.1.Event Listener|outline)

[12.4.2 Event Types 457](#12.4.2.Event Types|outline)

[12.5 Mathematical Content 458](#12.5.Mathematical Content|outline)

[12.6 DDE Connections 459](#12.6.DDE Connections|outline)

[12.6.1 Container for DDE Connection Declarations 459](#12.6.1.Container for DDE Connection Declarations|outline)

[12.6.2 Declaring DDE Connections for Text Fields 459](#12.6.2.Declaring DDE Connections for Text Fields|outline)

[12.6.3 Declaring DDE Connections for Tables 461](#12.6.3.Declaring DDE Connections for Tables |outline)

[13 SMIL Animations 463](#13.SMIL Animations|outline)

[13.1 Basic Animation Elements 463](#13.1.Basic Animation Elements|outline)

[13.1.1 Animate 463](#13.1.1.Animate|outline)

[13.1.2 Set 463](#13.1.2.Set|outline)

[13.1.3 Animate Motion 463](#13.1.3.Animate Motion|outline)

[13.1.4 Animate Color 464](#13.1.4.Animate Color|outline)

[13.1.5 Animate Transform 465](#13.1.5.Animate Transform|outline)

[13.1.6 Transition Filter 466](#13.1.6.Transition Filter|outline)

[13.2 Animation Model Attributes 467](#13.2.Animation Model Attributes|outline)

[13.3 Common Animation Attributes 467](#13.3.Common Animation Attributes|outline)

[13.3.1 Animation Target Attributes 468](#13.3.1.Animation Target Attributes|outline)

[13.3.2 Animation Function Attributes 468](#13.3.2.Animation Function Attributes|outline)

[13.4 Animation Timing 471](#13.4.Animation Timing|outline)

[13.4.1 Animation Timing Attributes 471](#13.4.1.Animation Timing Attributes|outline)

[13.4.2 Parallel Animations 475](#13.4.2.Parallel Animations|outline)

[13.4.3 Sequential Animations 475](#13.4.3.Sequential Animations|outline)

[13.4.4 Iterative Animations 475](#13.4.4.Iterative Animations|outline)

[13.5 Media Elements 476](#13.5.Media Elements|outline)

[13.5.1 Audio 476](#13.5.1.Audio|outline)

[13.6 Special Elements 477](#13.6.Special Elements|outline)

[13.6.1 Command 477](#13.6.1.Command|outline)

[14 Styles 479](#14.Styles|outline)

[14.1 Style Element 479](#14.1.Style Element|outline)

[14.1.1 Style Mappings 483](#14.1.1.Style Mappings|outline)

[14.2 Default Styles 485](#14.2.Default Styles|outline)

[14.3 Page Layout 485](#14.3.Page Layout|outline)

[14.3.1 Header and Footer Styles 486](#14.3.1.Header and Footer Styles|outline)

[14.4 Master Pages 487](#14.4.Master Pages|outline)

[14.4.1 Headers and Footers 489](#14.4.1.Headers and Footers|outline)

[14.4.2 Presentation Notes 491](#14.4.2.Presentation Notes|outline)

[14.5 Table Templates 493](#14.5.Table Templates|outline)

[14.5.1 Row and Column Styles 494](#14.5.1.Row and Column Styles|outline)

[14.6 Font Face Declaration 496](#14.6.Font Face Declaration|outline)

[14.6.1 CSS2/SVG Font Descriptors 496](#14.6.1.CSS2/SVG Font Descriptors|outline)

[14.6.2 Name 500](#14.6.2.Name|outline)

[14.6.3 Adornments 500](#14.6.3.Adornments|outline)

[14.6.4 Font Family Generic 500](#14.6.4.Font Family Generic|outline)

[14.6.5 Font Pitch 501](#14.6.5.Font Pitch|outline)

[14.6.6 Font Character Set 501](#14.6.6.Font Character Set|outline)

[14.7 Data Styles 501](#14.7.Data Styles|outline)

[14.7.1 Number Style 501](#14.7.1.Number Style|outline)

[14.7.2 Currency Style 505](#14.7.2.Currency Style|outline)

[14.7.3 Percentage Style 507](#14.7.3.Percentage Style|outline)

[14.7.4 Date Style 508](#14.7.4.Date Style|outline)

[14.7.5 Time Style 513](#14.7.5.Time Style|outline)

[14.7.6 Boolean Style 517](#14.7.6.Boolean Style|outline)

[14.7.7 Text Style 517](#14.7.7.Text Style|outline)

[14.7.8 Common Data Style Elements 518](#14.7.8.Common Data Style Elements|outline)

[14.7.9 Common Data Style Attributes 519](#14.7.9.Common Data Style Attributes|outline)

[14.7.10 Transliteration 522](#14.7.10.Transliteration|outline)

[14.7.11 Common Data Style Child Element Attributes 523](#14.7.11.Common Data Style Child Element Attributes|outline)

[14.8 Text Styles 525](#14.8.Text Styles|outline)

[14.8.1 Text Styles 525](#14.8.1.Text Styles|outline)

[14.8.2 Paragraph Styles 525](#14.8.2.Paragraph Styles|outline)

[14.8.3 Section Styles 526](#14.8.3.Section Styles|outline)

[14.8.4 Ruby Style 526](#14.8.4.Ruby Style|outline)

[14.9 Enhanced Text Styles 526](#14.9.Enhanced Text Styles|outline)

[14.9.1 Line Numbering Configuration 526](#14.9.1.Line Numbering Configuration|outline)

[14.9.2 Notes Configuration Element 530](#14.9.2.Notes Configuration Element|outline)

[14.9.3 Bibliography Configuration 533](#14.9.3.Bibliography Configuration|outline)

[14.10 List Style 535](#14.10.List Style|outline)

[14.10.1 Common List-Level Style Attributes 536](#14.10.1.Common List-Level Style Attributes|outline)

[14.10.2 Number Level Style 537](#14.10.2.Number Level Style|outline)

[14.10.3 Bullet Level Style 538](#14.10.3.Bullet Level Style|outline)

[14.10.4 Image Level Style 540](#14.10.4.Image Level Style|outline)

[14.10.5 List Level Style Example 541](#14.10.5.List Level Style Example|outline)

[14.11 Outline Style 541](#14.11.Outline Style|outline)

[14.11.1 Outline Level Style 542](#14.11.1.Outline Level Style|outline)

[14.12 Table Styles 543](#14.12.Table Styles|outline)

[14.12.1 Table Styles 543](#14.12.1.Table Styles|outline)

[14.12.2 Table Column Styles 544](#14.12.2.Table Column Styles|outline)

[14.12.3 Table Row Styles 544](#14.12.3.Table Row Styles|outline)

[14.12.4 Table Cell Styles 544](#14.12.4.Table Cell Styles|outline)

[14.13 Graphic Styles 545](#14.13.Graphic Styles|outline)

[14.13.1 Graphic and Presentation Styles 545](#14.13.1.Graphic and Presentation Styles|outline)

[14.13.2 Drawing Page Style 545](#14.13.2.Drawing Page Style|outline)

[14.14 Enhanced Graphic Style Elements 546](#14.14.Enhanced Graphic Style Elements|outline)

[14.14.1 Gradient 546](#14.14.1.Gradient|outline)

[14.14.2 SVG Gradients 549](#14.14.2.SVG Gradients|outline)

[14.14.3 Hatch 552](#14.14.3.Hatch|outline)

[14.14.4 Fill Image 554](#14.14.4.Fill Image|outline)

[14.14.5 Opacity Gradient 555](#14.14.5.Opacity Gradient|outline)

[14.14.6 Marker 556](#14.14.6.Marker|outline)

[14.14.7 Stroke Dash 557](#14.14.7.Stroke Dash|outline)

[14.15 Presentation Page Layouts 559](#14.15.Presentation Page Layouts|outline)

[14.15.1 Presentation Placeholder 559](#14.15.1.Presentation Placeholder|outline)

[14.16 Chart Styles 560](#14.16.Chart Styles|outline)

[15 Formatting Properties 561](#15.Formatting Properties|outline)

[15.1 Simple and Complex Formatting Properties 561](#15.1.Simple and Complex Formatting Properties|outline)

[15.1.1 Simple Formatting Properties 561](#15.1.1.Simple Formatting Properties|outline)

[15.1.2 Complex Formatting Properties 562](#15.1.2.Complex Formatting Properties|outline)

[15.1.3 Processing Rules for Formatting Properties 562](#15.1.3.Processing Rules for Formatting Properties|outline)

[15.2 Page Layout Formatting Properties 562](#15.2.Page Layout Formatting Properties|outline)

[15.2.1 Page Size 563](#15.2.1.Page Size|outline)

[15.2.2 Page Number Format 563](#15.2.2.Page Number Format|outline)

[15.2.3 Paper Tray 564](#15.2.3.Paper Tray|outline)

[15.2.4 Print Orientation 564](#15.2.4.Print Orientation|outline)

[15.2.5 Margins 564](#15.2.5.Margins|outline)

[15.2.6 Border 565](#15.2.6.Border|outline)

[15.2.7 Border Line Width 565](#15.2.7.Border Line Width|outline)

[15.2.8 Padding 565](#15.2.8.Padding|outline)

[15.2.9 Shadow 565](#15.2.9.Shadow|outline)

[15.2.10 Background 565](#15.2.10.Background|outline)

[15.2.11 Columns 566](#15.2.11.Columns|outline)

[15.2.12 Register-truth 566](#15.2.12.Register-truth|outline)

[15.2.13 Print 566](#15.2.13.Print|outline)

[15.2.14 Print Page Order 567](#15.2.14.Print Page Order|outline)

[15.2.15 First Page Number 567](#15.2.15.First Page Number|outline)

[15.2.16 Scale 567](#15.2.16.Scale|outline)

[15.2.17 Table Centering 568](#15.2.17.Table Centering|outline)

[15.2.18 Maximum Footnote Height 568](#15.2.18.Maximum Footnote Height|outline)

[15.2.19 Writing Mode 569](#15.2.19.Writing Mode|outline)

[15.2.20 Footnote Separator 569](#15.2.20.Footnote Separator|outline)

[15.2.21 Layout Grid 570](#15.2.21.Layout Grid|outline)

[15.2.22 Layout Grid Base Height 570](#15.2.22.Layout Grid Base Height|outline)

[15.2.23 Layout Grid Ruby Height 571](#15.2.23.Layout Grid Ruby Height|outline)

[15.2.24 Layout Grid Lines 571](#15.2.24.Layout Grid Lines|outline)

[15.2.25 Layout Grid Color 571](#15.2.25.Layout Grid Color|outline)

[15.2.26 Layout Grid Ruby Below 571](#15.2.26.Layout Grid Ruby Below|outline)

[15.2.27 Layout Grid Print 572](#15.2.27.Layout Grid Print|outline)

[15.2.28 Layout Grid Display 572](#15.2.28.Layout Grid Display|outline)

[15.3 Header Footer Formatting Properties 572](#15.3.Header Footer Formatting Properties|outline)

[15.3.1 Fixed and Minimum heights 573](#15.3.1.Fixed and Minimum heights|outline)

[15.3.2 Margins 573](#15.3.2.Margins|outline)

[15.3.3 Border 573](#15.3.3.Border|outline)

[15.3.4 Border Line Width 573](#15.3.4.Border Line Width|outline)

[15.3.5 Padding 574](#15.3.5.Padding|outline)

[15.3.6 Background 574](#15.3.6.Background|outline)

[15.3.7 Shadow 574](#15.3.7.Shadow|outline)

[15.3.8 Dynamic Spacing 574](#15.3.8.Dynamic Spacing|outline)

[15.4 Text Formatting Properties 575](#15.4.Text Formatting Properties|outline)

[15.4.1 Font Variant 575](#15.4.1.Font Variant|outline)

[15.4.2 Text Transformations 575](#15.4.2.Text Transformations|outline)

[15.4.3 Color 576](#15.4.3.Color|outline)

[15.4.4 Window Font Color 576](#15.4.4.Window Font Color|outline)

[15.4.5 Text Outline 576](#15.4.5.Text Outline|outline)

[15.4.6 Line-Through Type 576](#15.4.6.Line-Through Type|outline)

[15.4.7 Line-Through Style 577](#15.4.7.Line-Through Style|outline)

[15.4.8 Line-Through Width 577](#15.4.8.Line-Through Width|outline)

[15.4.9 Line-Through Color 577](#15.4.9.Line-Through Color|outline)

[15.4.10 Line-Through Text 578](#15.4.10.Line-Through Text|outline)

[15.4.11 Line-Through Text Style 578](#15.4.11.Line-Through Text Style|outline)

[15.4.12 Text Position 578](#15.4.12.Text Position|outline)

[15.4.13 Font Name 579](#15.4.13.Font Name|outline)

[15.4.14 Font Family 579](#15.4.14.Font Family|outline)

[15.4.15 Font Family Generic 580](#15.4.15.Font Family Generic|outline)

[15.4.16 Font Style 581](#15.4.16.Font Style|outline)

[15.4.17 Font Pitch 581](#15.4.17.Font Pitch|outline)

[15.4.18 Font Character Set 582](#15.4.18.Font Character Set|outline)

[15.4.19 Font Size 582](#15.4.19.Font Size|outline)

[15.4.20 Relative Font Size 583](#15.4.20.Relative Font Size|outline)

[15.4.21 Script Type 584](#15.4.21.Script Type|outline)

[15.4.22 Letter Spacing 584](#15.4.22.Letter Spacing|outline)

[15.4.23 Language 584](#15.4.23.Language|outline)

[15.4.24 Country 585](#15.4.24.Country|outline)

[15.4.25 Font Style 585](#15.4.25.Font Style|outline)

[15.4.26 Font Relief 586](#15.4.26.Font Relief|outline)

[15.4.27 Text Shadow 586](#15.4.27.Text Shadow|outline)

[15.4.28 Underlining Type 587](#15.4.28.Underlining Type|outline)

[15.4.29 Underlining Style 587](#15.4.29.Underlining Style|outline)

[15.4.30 Underling Width 588](#15.4.30.Underling Width|outline)

[15.4.31 Underline Color 588](#15.4.31.Underline Color|outline)

[15.4.32 Font Weight 588](#15.4.32.Font Weight|outline)

[15.4.33 Text Underline Word Mode 589](#15.4.33.Text Underline Word Mode|outline)

[15.4.34 Text Line-Through Word Mode 589](#15.4.34.Text Line-Through Word Mode|outline)

[15.4.35 Letter Kerning 590](#15.4.35.Letter Kerning|outline)

[15.4.36 Text Blinking 590](#15.4.36.Text Blinking|outline)

[15.4.37 Text Background Color 590](#15.4.37.Text Background Color|outline)

[15.4.38 Text Combine 590](#15.4.38.Text Combine|outline)

[15.4.39 Text Combine Start and End Characters 591](#15.4.39.Text Combine Start and End Characters|outline)

[15.4.40 Text Emphasis 591](#15.4.40.Text Emphasis|outline)

[15.4.41 Text Scale 592](#15.4.41.Text Scale|outline)

[15.4.42 Text Rotation Angle 592](#15.4.42.Text Rotation Angle|outline)

[15.4.43 Text Rotation Scale 592](#15.4.43.Text Rotation Scale|outline)

[15.4.44 Hyphenation 592](#15.4.44.Hyphenation|outline)

[15.4.45 Hyphenation Remain Char Count 593](#15.4.45.Hyphenation Remain Char Count|outline)

[15.4.46 Hyphenation Push Char Count 593](#15.4.46.Hyphenation Push Char Count|outline)

[15.4.47 Hidden or Conditional Text 593](#15.4.47.Hidden or Conditional Text|outline)

[15.5 Paragraph Formatting Properties 594](#15.5.Paragraph Formatting Properties|outline)

[15.5.1 Fixed Line Height 594](#15.5.1.Fixed Line Height|outline)

[15.5.2 Minimum Line Height 595](#15.5.2.Minimum Line Height|outline)

[15.5.3 Line Distance 595](#15.5.3.Line Distance|outline)

[15.5.4 Font-Independent Line Spacing 595](#15.5.4.Font-Independent Line Spacing|outline)

[15.5.5 Text Align 595](#15.5.5.Text Align|outline)

[15.5.6 Text Align of Last Line 596](#15.5.6.Text Align of Last Line|outline)

[15.5.7 Justify Single Word 596](#15.5.7.Justify Single Word|outline)

[15.5.8 Keep Together 597](#15.5.8.Keep Together|outline)

[15.5.9 Widows 597](#15.5.9.Widows|outline)

[15.5.10 Orphans 597](#15.5.10.Orphans|outline)

[15.5.11 Tab Stops 597](#15.5.11.Tab Stops|outline)

[15.5.12 Tab Stop Distance 601](#15.5.12.Tab Stop Distance|outline)

[15.5.13 Hyphenation Keep 601](#15.5.13.Hyphenation Keep|outline)

[15.5.14 Maximum Hyphens 601](#15.5.14.Maximum Hyphens|outline)

[15.5.15 Drop Caps 602](#15.5.15.Drop Caps|outline)

[15.5.16 Register True 603](#15.5.16.Register True |outline)

[15.5.17 Left and Right Margins 603](#15.5.17.Left and Right Margins|outline)

[15.5.18 Text Indent 604](#15.5.18.Text Indent|outline)

[15.5.19 Automatic Text Indent 604](#15.5.19.Automatic Text Indent|outline)

[15.5.20 Top and Bottom Margins 605](#15.5.20.Top and Bottom Margins|outline)

[15.5.21 Margins 605](#15.5.21.Margins|outline)

[15.5.22 Break Before and Break After 605](#15.5.22.Break Before and Break After|outline)

[15.5.23 Paragraph Background Color 606](#15.5.23.Paragraph Background Color|outline)

[15.5.24 Paragraph Background Image 606](#15.5.24.Paragraph Background Image|outline)

[15.5.25 Border 609](#15.5.25.Border|outline)

[15.5.26 Border Line Width 609](#15.5.26.Border Line Width|outline)

[15.5.27 Padding 610](#15.5.27.Padding|outline)

[15.5.28 Shadow 611](#15.5.28.Shadow|outline)

[15.5.29 Keep with Next 612](#15.5.29.Keep with Next|outline)

[15.5.30 Line Numbering 612](#15.5.30.Line Numbering|outline)

[15.5.31 Line Number Start Value 612](#15.5.31.Line Number Start Value|outline)

[15.5.32 Text Autospace 612](#15.5.32.Text Autospace|outline)

[15.5.33 Punctuation Wrap 613](#15.5.33.Punctuation Wrap|outline)

[15.5.34 Line Break 613](#15.5.34.Line Break|outline)

[15.5.35 Vertical Alignment 613](#15.5.35.Vertical Alignment|outline)

[15.5.36 Writing Mode 614](#15.5.36.Writing Mode|outline)

[15.5.37 Automatic Writing Mode 614](#15.5.37.Automatic Writing Mode|outline)

[15.5.38 Snap To Layout 615](#15.5.38.Snap To Layout|outline)

[15.5.39 Page Number 615](#15.5.39.Page Number|outline)

[15.5.40 Background Transparency 615](#15.5.40.Background Transparency|outline)

[15.6 Ruby Text Formatting Properties 615](#15.6.Ruby Text Formatting Properties|outline)

[15.6.1 Ruby Position 616](#15.6.1.Ruby Position|outline)

[15.6.2 Ruby Alignment 616](#15.6.2.Ruby Alignment|outline)

[15.7 Section Formatting Properties 616](#15.7.Section Formatting Properties|outline)

[15.7.1 Section Background 617](#15.7.1.Section Background|outline)

[15.7.2 Margins 617](#15.7.2.Margins|outline)

[15.7.3 Columns 617](#15.7.3.Columns|outline)

[15.7.4 Column Specification 618](#15.7.4.Column Specification|outline)

[15.7.5 Column Separator 620](#15.7.5.Column Separator|outline)

[15.7.6 Protect 621](#15.7.6.Protect|outline)

[15.7.7 Don't Balance Text Columns 621](#15.7.7.Don't Balance Text Columns|outline)

[15.7.8 Writing Mode 622](#15.7.8.Writing Mode|outline)

[15.7.9 Notes Configuration 622](#15.7.9.Notes Configuration|outline)

[15.8 Table Formatting Properties 622](#15.8.Table Formatting Properties|outline)

[15.8.1 Table Width 622](#15.8.1.Table Width|outline)

[15.8.2 Table Alignment 623](#15.8.2.Table Alignment|outline)

[15.8.3 Table Left and Right Margin 623](#15.8.3.Table Left and Right Margin|outline)

[15.8.4 Table Top and Bottom Margin 624](#15.8.4.Table Top and Bottom Margin|outline)

[15.8.5 Table Margins 624](#15.8.5.Table Margins|outline)

[15.8.6 Page Number 624](#15.8.6.Page Number|outline)

[15.8.7 Break Before and Break After 624](#15.8.7.Break Before and Break After|outline)

[15.8.8 Table Background and Background Image 624](#15.8.8.Table Background and Background Image|outline)

[15.8.9 Table Shadow 624](#15.8.9.Table Shadow|outline)

[15.8.10 Keep with Next 625](#15.8.10.Keep with Next|outline)

[15.8.11 May Break Between Rows 625](#15.8.11.May Break Between Rows|outline)

[15.8.12 Border Model Property 625](#15.8.12.Border Model Property|outline)

[15.8.13 Writing Mode 626](#15.8.13.Writing Mode|outline)

[15.8.14 Display 626](#15.8.14.Display|outline)

[15.9 Column Formatting Properties 626](#15.9.Column Formatting Properties|outline)

[15.9.1 Column Width 626](#15.9.1.Column Width|outline)

[15.9.2 Optimal Table Column Width 627](#15.9.2.Optimal Table Column Width|outline)

[15.9.3 Break Before and Break After 627](#15.9.3.Break Before and Break After|outline)

[15.10 Table Row Formatting Properties 627](#15.10.Table Row Formatting Properties|outline)

[15.10.1 Row Height 627](#15.10.1.Row Height|outline)

[15.10.2 Optimal Table Row Height 628](#15.10.2.Optimal Table Row Height|outline)

[15.10.3 Row Background 628](#15.10.3.Row Background|outline)

[15.10.4 Break Before and Break After 628](#15.10.4.Break Before and Break After|outline)

[15.10.5 Keep Together 628](#15.10.5.Keep Together|outline)

[15.11 Table Cell Formatting Properties 629](#15.11.Table Cell Formatting Properties|outline)

[15.11.1 Vertical Alignment 629](#15.11.1.Vertical Alignment|outline)

[15.11.2 Text Align Source 629](#15.11.2.Text Align Source|outline)

[15.11.3 Direction 630](#15.11.3.Direction|outline)

[15.11.4 Vertical Glyph Orientation 630](#15.11.4.Vertical Glyph Orientation|outline)

[15.11.5 Cell Shadow 630](#15.11.5.Cell Shadow|outline)

[15.11.6 Cell Background 631](#15.11.6.Cell Background|outline)

[15.11.7 Cell Border 631](#15.11.7.Cell Border|outline)

[15.11.8 Diagonal Lines 631](#15.11.8.Diagonal Lines|outline)

[15.11.9 Border Line Width 632](#15.11.9.Border Line Width|outline)

[15.11.10 Padding 632](#15.11.10.Padding|outline)

[15.11.11 Wrap Option 632](#15.11.11.Wrap Option|outline)

[15.11.12 Rotation Angle 632](#15.11.12.Rotation Angle|outline)

[15.11.13 Rotation Align 633](#15.11.13.Rotation Align|outline)

[15.11.14 Cell Protect 633](#15.11.14.Cell Protect|outline)

[15.11.15 Print Content 633](#15.11.15.Print Content|outline)

[15.11.16 Decimal places 634](#15.11.16.Decimal places|outline)

[15.11.17 Repeat Content 634](#15.11.17.Repeat Content|outline)

[15.11.18 Shrink To Fit 634](#15.11.18.Shrink To Fit|outline)

[15.12 List-Level Style Properties 635](#15.12.List-Level Style Properties|outline)

[15.13 Stroke Properties 637](#15.13.Stroke Properties|outline)

[15.13.1 Stroke Style 637](#15.13.1.Stroke Style|outline)

[15.13.2 Dash 638](#15.13.2.Dash|outline)

[15.13.3 Multiple Dashes 638](#15.13.3.Multiple Dashes|outline)

[15.13.4 Width 638](#15.13.4.Width|outline)

[15.13.5 Color 638](#15.13.5.Color|outline)

[15.13.6 Start Marker 638](#15.13.6.Start Marker|outline)

[15.13.7 End Marker 639](#15.13.7.End Marker|outline)

[15.13.8 Start Marker Width 639](#15.13.8.Start Marker Width|outline)

[15.13.9 End Marker Width 639](#15.13.9.End Marker Width|outline)

[15.13.10 Start Marker Center 639](#15.13.10.Start Marker Center|outline)

[15.13.11 End Marker Center 640](#15.13.11.End Marker Center|outline)

[15.13.12 Opacity 640](#15.13.12.Opacity|outline)

[15.13.13 Line Join 640](#15.13.13.Line Join|outline)

[15.14 Fill Properties 640](#15.14.Fill Properties|outline)

[15.14.1 Fill Style 641](#15.14.1.Fill Style|outline)

[15.14.2 Color 641](#15.14.2.Color|outline)

[15.14.3 Secondary Fill Color 642](#15.14.3.Secondary Fill Color|outline)

[15.14.4 Gradient 642](#15.14.4.Gradient|outline)

[15.14.5 Gradient Step Count 642](#15.14.5.Gradient Step Count|outline)

[15.14.6 Hatch 642](#15.14.6.Hatch|outline)

[15.14.7 Solid Hatch 643](#15.14.7.Solid Hatch|outline)

[15.14.8 Fill Image 643](#15.14.8.Fill Image|outline)

[15.14.9 Fill Image Rendering Style 643](#15.14.9.Fill Image Rendering Style|outline)

[15.14.10 Fill Image Size 643](#15.14.10.Fill Image Size|outline)

[15.14.11 Fill Image Tile Reference Point 644](#15.14.11.Fill Image Tile Reference Point|outline)

[15.14.12 Fill Image Tile Translation 645](#15.14.12.Fill Image Tile Translation|outline)

[15.14.13 None and Linear Opacity 645](#15.14.13.None and Linear Opacity|outline)

[15.14.14 Gradient Opacity 645](#15.14.14.Gradient Opacity|outline)

[15.14.15 Fill Rule 645](#15.14.15.Fill Rule|outline)

[15.14.16 Symbol color 646](#15.14.16.Symbol color|outline)

[15.15 Text Animation Properties 646](#15.15.Text Animation Properties|outline)

[15.15.1 Animation 646](#15.15.1.Animation|outline)

[15.15.2 Animation Direction 647](#15.15.2.Animation Direction|outline)

[15.15.3 Animation Start Inside 647](#15.15.3.Animation Start Inside|outline)

[15.15.4 Animation Stop Inside 647](#15.15.4.Animation Stop Inside|outline)

[15.15.5 Animation Repeat 647](#15.15.5.Animation Repeat|outline)

[15.15.6 Animation Delay 648](#15.15.6.Animation Delay|outline)

[15.15.7 Animation Steps 648](#15.15.7.Animation Steps|outline)

[15.16 Text and Text Alignment Properties 648](#15.16.Text and Text Alignment Properties|outline)

[15.16.1 Auto Grow Width and Height 648](#15.16.1.Auto Grow Width and Height|outline)

[15.16.2 Fit To Size 649](#15.16.2.Fit To Size|outline)

[15.16.3 Fit To Contour 649](#15.16.3.Fit To Contour|outline)

[15.16.4 Text Area Vertical Align 649](#15.16.4.Text Area Vertical Align|outline)

[15.16.5 Text Area Horizontal Align 649](#15.16.5.Text Area Horizontal Align|outline)

[15.16.6 Word Wrap 650](#15.16.6.Word Wrap|outline)

[15.16.7 List Styles 650](#15.16.7.List Styles|outline)

[15.17 Color Properties 650](#15.17.Color Properties|outline)

[15.17.1 Color Mode 650](#15.17.1.Color Mode |outline)

[15.17.2 Color Inversion 651](#15.17.2.Color Inversion|outline)

[15.17.3 Adjust Luminance 651](#15.17.3.Adjust Luminance|outline)

[15.17.4 Adjust Contrast 651](#15.17.4.Adjust Contrast|outline)

[15.17.5 Adjust Gamma 651](#15.17.5.Adjust Gamma|outline)

[15.17.6 Adjust Red 652](#15.17.6.Adjust Red|outline)

[15.17.7 Adjust Green 652](#15.17.7.Adjust Green|outline)

[15.17.8 Adjust Blue 652](#15.17.8.Adjust Blue|outline)

[15.17.9 Adjust Opacity 652](#15.17.9.Adjust Opacity|outline)

[15.18 Shadow Properties 652](#15.18.Shadow Properties|outline)

[15.18.1 Shadow 653](#15.18.1.Shadow|outline)

[15.18.2 Offset 653](#15.18.2.Offset|outline)

[15.18.3 Color 653](#15.18.3.Color|outline)

[15.18.4 Opacity 653](#15.18.4.Opacity|outline)

[15.19 Connector Properties 654](#15.19.Connector Properties|outline)

[15.19.1 Start Line Spacing 654](#15.19.1.Start Line Spacing|outline)

[15.19.2 End Line Spacing 654](#15.19.2.End Line Spacing|outline)

[15.20 Measure Properties 654](#15.20.Measure Properties|outline)

[15.20.1 Line Distance 654](#15.20.1.Line Distance|outline)

[15.20.2 Guide Overhang 655](#15.20.2.Guide Overhang|outline)

[15.20.3 Guide Distance 655](#15.20.3.Guide Distance|outline)

[15.20.4 Start Guide 655](#15.20.4.Start Guide|outline)

[15.20.5 End Guide 655](#15.20.5.End Guide|outline)

[15.20.6 Placing 656](#15.20.6.Placing|outline)

[15.20.7 Parallel 656](#15.20.7.Parallel|outline)

[15.20.8 Text Alignment 656](#15.20.8.Text Alignment|outline)

[15.20.9 Unit 657](#15.20.9.Unit|outline)

[15.20.10 Show Unit 657](#15.20.10.Show Unit|outline)

[15.20.11 Decimal Places 657](#15.20.11.Decimal Places|outline)

[15.21 Caption Properties 657](#15.21.Caption Properties|outline)

[15.21.1 Type 658](#15.21.1.Type|outline)

[15.21.2 Angle Type 658](#15.21.2.Angle Type|outline)

[15.21.3 Angle 658](#15.21.3.Angle|outline)

[15.21.4 Gap 659](#15.21.4.Gap|outline)

[15.21.5 Escape Direction 659](#15.21.5.Escape Direction|outline)

[15.21.6 Escape 659](#15.21.6.Escape|outline)

[15.21.7 Line Length 660](#15.21.7.Line Length|outline)

[15.21.8 Fit Line Length 660](#15.21.8.Fit Line Length|outline)

[15.22 3D Geometry Properties 660](#15.22.3D Geometry Properties|outline)

[15.22.1 Horizontal Segments 660](#15.22.1.Horizontal Segments|outline)

[15.22.2 Vertical Segments 660](#15.22.2.Vertical Segments|outline)

[15.22.3 Edge Rounding 661](#15.22.3.Edge Rounding|outline)

[15.22.4 Edge Rounding Mode 661](#15.22.4.Edge Rounding Mode|outline)

[15.22.5 Back Scale 661](#15.22.5.Back Scale|outline)

[15.22.6 Depth 661](#15.22.6.Depth|outline)

[15.22.7 Backface Culling 662](#15.22.7.Backface Culling|outline)

[15.22.8 End Angle 662](#15.22.8.End Angle|outline)

[15.22.9 Close Front 662](#15.22.9.Close Front|outline)

[15.22.10 Close Back 662](#15.22.10.Close Back|outline)

[15.23 3D Lighting Properties 663](#15.23.3D Lighting Properties|outline)

[15.23.1 Mode 663](#15.23.1.Mode|outline)

[15.23.2 Normals Kind 663](#15.23.2.Normals Kind|outline)

[15.23.3 Normals Direction 663](#15.23.3.Normals Direction|outline)

[15.24 3D Texture Properties 664](#15.24.3D Texture Properties|outline)

[15.24.1 Generation Mode 664](#15.24.1.Generation Mode|outline)

[15.24.2 Kind 664](#15.24.2.Kind|outline)

[15.24.3 Filter 665](#15.24.3.Filter|outline)

[15.24.4 Mode 665](#15.24.4.Mode|outline)

[15.25 3D Material Properties 665](#15.25.3D Material Properties|outline)

[15.25.1 Colors 665](#15.25.1.Colors|outline)

[15.25.2 Shininess 666](#15.25.2.Shininess|outline)

[15.26 3D Shadow Properties 666](#15.26.3D Shadow Properties|outline)

[15.26.1 Shadow 666](#15.26.1.Shadow|outline)

[15.27 Frame Formatting Properties 666](#15.27.Frame Formatting Properties|outline)

[15.27.1 Frame Widths 666](#15.27.1.Frame Widths|outline)

[15.27.2 Frame Heights 667](#15.27.2.Frame Heights|outline)

[15.27.3 Maximum Width and Height 667](#15.27.3.Maximum Width and Height|outline)

[15.27.4 Left and Right Margins 668](#15.27.4.Left and Right Margins|outline)

[15.27.5 Top and Bottom Margins 668](#15.27.5.Top and Bottom Margins|outline)

[15.27.6 Margins 668](#15.27.6.Margins|outline)

[15.27.7 Print Content 668](#15.27.7.Print Content|outline)

[15.27.8 Protect 668](#15.27.8.Protect|outline)

[15.27.9 Horizontal Position 669](#15.27.9.Horizontal Position|outline)

[15.27.10 Horizontal Relation 670](#15.27.10.Horizontal Relation|outline)

[15.27.11 Vertical Position 671](#15.27.11.Vertical Position|outline)

[15.27.12 Vertical Relation 672](#15.27.12.Vertical Relation|outline)

[15.27.13 Frame Anchor 673](#15.27.13.Frame Anchor|outline)

[15.27.14 Border 673](#15.27.14.Border|outline)

[15.27.15 Border Line Width 673](#15.27.15.Border Line Width|outline)

[15.27.16 Padding 673](#15.27.16.Padding|outline)

[15.27.17 Shadow 674](#15.27.17.Shadow|outline)

[15.27.18 Background 674](#15.27.18.Background|outline)

[15.27.19 Columns 674](#15.27.19.Columns|outline)

[15.27.20 Editable 674](#15.27.20.Editable|outline)

[15.27.21 Wrapping 674](#15.27.21.Wrapping|outline)

[15.27.22 Dynamic Wrap Threshold 675](#15.27.22.Dynamic Wrap Threshold|outline)

[15.27.23 Paragraph-only Wrapping 675](#15.27.23.Paragraph-only Wrapping|outline)

[15.27.24 Contour Wrapping 676](#15.27.24.Contour Wrapping|outline)

[15.27.25 Contour Wrapping Mode 676](#15.27.25.Contour Wrapping Mode|outline)

[15.27.26 Run Through 676](#15.27.26.Run Through|outline)

[15.27.27 Flow with Text 677](#15.27.27.Flow with Text|outline)

[15.27.28 Overflow behavior 677](#15.27.28.Overflow behavior|outline)

[15.27.29 Mirroring 677](#15.27.29.Mirroring|outline)

[15.27.30 Clipping 678](#15.27.30.Clipping|outline)

[15.27.31 Wrap Influence on Position 678](#15.27.31.Wrap Influence on Position|outline)

[15.27.32 Writing Mode 679](#15.27.32.Writing Mode|outline)

[15.28 Floating Frame Formatting Properties 679](#15.28.Floating Frame Formatting Properties|outline)

[15.28.1 Display Scrollbar 679](#15.28.1.Display Scrollbar|outline)

[15.28.2 Display Border 680](#15.28.2.Display Border|outline)

[15.28.3 Margins 680](#15.28.3.Margins|outline)

[15.28.4 Object Formatting Properties 680](#15.28.4.Object Formatting Properties|outline)

[15.28.5 Visible Area 680](#15.28.5.Visible Area|outline)

[15.28.6 Draw Aspect 681](#15.28.6.Draw Aspect|outline)

[15.29 Chart Formatting Properties 681](#15.29.Chart Formatting Properties|outline)

[15.29.1 Scale Text 682](#15.29.1.Scale Text|outline)

[15.30 Chart Subtype Properties 682](#15.30.Chart Subtype Properties|outline)

[15.30.1 Three-dimensional Charts 682](#15.30.1.Three-dimensional Charts|outline)

[15.30.2 Chart Depth 682](#15.30.2.Chart Depth|outline)

[15.30.3 Chart Symbol 683](#15.30.3.Chart Symbol|outline)

[15.30.4 Chart Symbol Size 683](#15.30.4.Chart Symbol Size|outline)

[15.30.5 Bar Chart Properties 684](#15.30.5.Bar Chart Properties|outline)

[15.30.6 Stock Chart Properties 684](#15.30.6.Stock Chart Properties|outline)

[15.30.7 Line Chart Properties 685](#15.30.7.Line Chart Properties|outline)

[15.30.8 Pie Chart Properties 686](#15.30.8.Pie Chart Properties|outline)

[15.30.9 Lines 686](#15.30.9.Lines|outline)

[15.30.10 Solid Charts Bars 686](#15.30.10.Solid Charts Bars|outline)

[15.30.11 Stacked Chart Bars 686](#15.30.11.Stacked Chart Bars|outline)

[15.31 Chart Axes Properties 687](#15.31.Chart Axes Properties|outline)

[15.31.1 Linked Data Formats 687](#15.31.1.Linked Data Formats|outline)

[15.31.2 Visibility 687](#15.31.2.Visibility|outline)

[15.31.3 Scaling 687](#15.31.3.Scaling|outline)

[15.31.4 Tick Marks 688](#15.31.4.Tick Marks|outline)

[15.31.5 Labels 689](#15.31.5.Labels|outline)

[15.32 Common Chart Properties 689](#15.32.Common Chart Properties|outline)

[15.32.1 Stacked Text 690](#15.32.1.Stacked Text|outline)

[15.32.2 Rotation Angle 690](#15.32.2.Rotation Angle|outline)

[15.32.3 Data Labels 690](#15.32.3.Data Labels|outline)

[15.33 Statistical Properties 691](#15.33.Statistical Properties|outline)

[15.33.1 Mean Value 691](#15.33.1.Mean Value|outline)

[15.33.2 Error Category 691](#15.33.2.Error Category|outline)

[15.34 Plot Area Properties 693](#15.34.Plot Area Properties|outline)

[15.34.1 Series Source 693](#15.34.1.Series Source|outline)

[15.35 Regression Curve Properties 693](#15.35.Regression Curve Properties|outline)

[15.35.1 Regression Type 694](#15.35.1.Regression Type|outline)

[15.36 Presentation Page Attributes 694](#15.36.Presentation Page Attributes|outline)

[15.36.1 Transition Type 695](#15.36.1.Transition Type|outline)

[15.36.2 Transition Style 695](#15.36.2.Transition Style|outline)

[15.36.3 Transition Speed 697](#15.36.3.Transition Speed|outline)

[15.36.4 Transition Type or Family 698](#15.36.4.Transition Type or Family|outline)

[15.36.5 Transition Subtype 698](#15.36.5.Transition Subtype|outline)

[15.36.6 Transition Direction 698](#15.36.6.Transition Direction|outline)

[15.36.7 Fade Color 698](#15.36.7.Fade Color|outline)

[15.36.8 Page Duration 699](#15.36.8.Page Duration|outline)

[15.36.9 Page Visibility 699](#15.36.9.Page Visibility|outline)

[15.36.10 Sound 699](#15.36.10.Sound|outline)

[15.36.11 Background Size 699](#15.36.11.Background Size|outline)

[15.36.12 Background Objects Visible 700](#15.36.12.Background Objects Visible|outline)

[15.36.13 Background Visible 700](#15.36.13.Background Visible|outline)

[15.36.14 Display Header 700](#15.36.14.Display Header|outline)

[15.36.15 Display Footer 700](#15.36.15.Display Footer|outline)

[15.36.16 Display Page Number 701](#15.36.16.Display Page Number|outline)

[15.36.17 Display Date And Time 701](#15.36.17.Display Date And Time|outline)

[16 Data Types and Schema Definitions 702](#16.Data Types and Schema Definitions|outline)

[16.1 Data Types 702](#16.1.Data Types|outline)

[16.2 Other Definitions 707](#16.2.Other Definitions|outline)

[16.3 Relax-NG Schema Suffix 708](#16.3.Relax-NG Schema Suffix|outline)

[17 Packages 709](#17.Packages|outline)

[17.1 Introduction 709](#17.1.Introduction|outline)

[17.2 Zip File Structure 709](#17.2.Zip File Structure|outline)

[17.3 Encryption 710](#17.3.Encryption|outline)

[17.4 MIME Type Stream 710](#17.4.MIME Type Stream|outline)

[17.5 Usage of IRIs Within Packages 711](#17.5.Usage of IRIs Within Packages|outline)

[17.6 Preview Image 711](#17.6.Preview Image|outline)

[17.7 Manifest File 711](#17.7.Manifest File|outline)

[17.7.1 Relax-NG Schema 712](#17.7.1.Relax-NG Schema|outline)

[17.7.2 Manifest Root Element 712](#17.7.2.Manifest Root Element|outline)

[17.7.3 File Entry 712](#17.7.3.File Entry|outline)

[17.7.4 Encryption Data 713](#17.7.4.Encryption Data|outline)

[17.7.5 Algorithm 714](#17.7.5.Algorithm|outline)

[17.7.6 Key Derivation 715](#17.7.6.Key Derivation|outline)

[17.7.7 Relax-NG Schema Suffix 717](#17.7.7.Relax-NG Schema Suffix|outline)

Appendix A. Strict Relax NG Schema 718

Appendix B. References 720

Appendix C. MIME Types and File Name Extensions (Non Normative) 722

Appendix D. Core Features Sets (Non Normative) 724

Appendix E. Accessibility Guidelines (Non Normative) 729

E.1. Title, Description and Caption of Graphical Elements 729

E.2. Hyperlink Titles 729

E.3. Tables in Presentations 730

E.4. Further Guidelines 730

Appendix F. Bidirectional (BiDi) Scripts,Numeric Digits Presentation and Calendars (Non Normative) 731

Appendix G. Changes From Previous Specification Versions (Non Normative) 733

G.1. Changes from “Open Office Specification 1.0 Committee Draft 1” 733

G.2. Changes from “Open Document Format for Office Applications (OpenDocument) 1.0 Committee Draft 2” 733

G.3. Changes from “Open Document Format for Office Applications (OpenDocument) v1.0” 734

G.4. Changes from “Open Document Format for Office Applications (OpenDocument) v1.0 (Second Edition)” 734

Appendix H. Acknowledgments (Non Normative) 737

1. Introduction

## Introduction

This document defines an XML schema for office applications and its semantics. The schema is suitable for office documents, including text documents, spreadsheets, charts and graphical documents like drawings or presentations, but is not restricted to these kinds of documents.

The schema provides for high-level information suitable for editing documents. It defines suitable XML structures for office documents and is friendly to transformations using XSLT or similar XML-based tools.

Chapter 1 contains the introduction to the OpenDocument format. The structure of documents that conform to the OpenDocument specification is explained in chapter 2. Chapter 3 described the meta information that can be contained in such documents. Chapters 4 and 5 describe their text and paragraph content. Text Fields are described in chapter 6, text indices in chapter 7.

Chapter 8 describes the table content of a document in OpenDocument format, chapter 9 its graphical content, chapter 10 its chart content, and chapter 11 its form content. Content that is common to all documents is described in chapter 12. The integration of SMIL animation markup into the OpenDocument schema is described in chapter 13. Chapter 14 explains style information content, chapter 15 specifies formatting properties that are can be used within styles. The data types used by the OpenDocument schema are described in chapter 16.

The OpenDocument format makes use of a package concept. These packages are described in chapter 17.

## Notation

Within this specification, the key words "shall", "shall not", "should", "should not" and "may" are to be interpreted as described in Annex H of [ISO/IEC Directives] if they appear in bold letters.

## Namespaces

Table 1 lists the namespaces that are defined by the OpenDocument format and their default prefixes. For more information about XML namespaces, please refer to the Namespaces in XML specification [xml-names].

Table 1: XML Namespaces defined by the OpenDocument schema

| Prefix | Description | Namespace |
| --- | --- | --- |
| office | For all common pieces of information that are not contained in another, more specific namespace. | urn:oasis:names:tc:opendocument:xmlns: office:1.0 |
| meta | For elements and attributes that describe meta information. | urn:oasis:names:tc:opendocument:xmlns: meta:1.0 |
| config | For elements and attributes that describe application specific settings. | urn:oasis:names:tc:opendocument:xmlns: config:1.0 |
| text | For elements and attributes that may occur within text documents and text parts of other document types, such as the contents of a spreadsheet cell. | urn:oasis:names:tc:opendocument:xmlns: text:1.0 |
| table | For elements and attributes that may occur within spreadsheets or within table definitions of a text document. | urn:oasis:names:tc:opendocument:xmlns: table:1.0 |
| drawing | For elements and attributes that describe graphic content. | urn:oasis:names:tc:opendocument:xmlns: drawing:1.0 |
| presentation | For elements and attributes that describe presentation content. | urn:oasis:names:tc:opendocument:xmlns: presentation:1.0 |
| dr3d | For elements and attributes that describe 3D graphic content. | urn:oasis:names:tc:opendocument:xmlns: dr3d:1.0 |
| anim | For elements and attributes that describe animation content. | urn:oasis:names:tc:opendocument:xmlns: animation:1.0 |
| chart | For elements and attributes that describe chart content. | urn:oasis:names:tc:opendocument:xmlns: chart:1.0 |
| form | For elements and attributes that describe forms and controls. | urn:oasis:names:tc:opendocument:xmlns: form:1.0 |
| script | For elements and attributes that represent scripts or events. | urn:oasis:names:tc:opendocument:xmlns: script:1.0 |
| style | For elements and attributes that describe the style and inheritance model used by the OpenDocument format as well as some common formatting attributes. | urn:oasis:names:tc:opendocument:xmlns: style:1.0 |
| number | For elements and attributes that describe data style information. | urn:oasis:names:tc:opendocument:xmlns: data style:1.0 |
| manifest | For elements and attribute contained in the package manifest. | urn:oasis:names:tc:opendocument:xmlns: manifest:1.0 |

Table 2 lists the namespaces that are defined by the OpenDocument format, but contain elements and attributes whose semantics are compatible to elements and attributes from other specifications.

Table 2: XML Namespaces defined by the OpenDocument schema that include elements and attributes that are compatible to elements and attributes of other standards.

| Prefix | Description | Namespace |
| --- | --- | --- |
| fo | For attributes that are compatible to attributes defined in [XSL]. | urn:oasis:names:tc:opendocument:xmlns: xsl-fo-compatible:1.0 |
| svg | For elements and attributes that are compatible to elements or attributes defined in [SVG]. | urn:oasis:names:tc:opendocument:xmlns: svg-compatible:1.0 |
| smil | For attributes that are compatible to attributes defined in [SMIL20]. | urn:oasis:names:tc:opendocument:xmlns: smil-compatible:1.0 |

Table 3 lists the namespaces that are imported into the OpenDocument format and their default prefixes.

Table 3: XML Namespaces used by the OpenDocument schema

| Prefix | Description | Namespace |
| --- | --- | --- |
| dc | The Dublin Core Namespace (see [DCMI]). | http://purl.org/dc/elements/1.1/ |
| xlink | The XLink namespace (see [XLink]). | http://www.w3.org/1999/xlink |
| math | MathML Namespace (see [MathML]) | http://www.w3.org/1998/Math/MathML |
| xforms | The XForms namespace (see [XForms]). | http://www.w3.org/2002/xforms |

## Relax-NG Schema

The normative XML Schema for the OpenDocument format is embedded within this specification. It can be obtained from the specification document by concatenating all schema fragments contained in chapters 1 to 16. All schema fragments have a gray background color and line numbers.

The schema language used within this specification is Relax-NG (see [RNG]). The attribute default value feature specified in [RNG-Compat] is used to provide attribute default values.

The schema provided in this specification permits arbitrary content within meta information elements and formatting properties elements as described in section 1.5. Appendix A contains a schema that restricts the content within these elements to the attributes and elements defined in this specification.

Prefix for the normative Relax-NG schema:

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

<!--

OASIS OpenDocument v1.1

OASIS Standard, 1 Feb 2007

Relax-NG Schema

$Id$

© 2002-2007 OASIS Open

© 1999-2007 Sun Microsystems, Inc.

-->

<grammar

xmlns="http://relaxng.org/ns/structure/1.0"

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

datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes"

xmlns:office="urn:oasis:names:tc:opendocument:xmlns:office:1.0"

xmlns:meta="urn:oasis:names:tc:opendocument:xmlns:meta:1.0"

xmlns:config="urn:oasis:names:tc:opendocument:xmlns:config:1.0"

xmlns:text="urn:oasis:names:tc:opendocument:xmlns:text:1.0"

xmlns:table="urn:oasis:names:tc:opendocument:xmlns:table:1.0"

xmlns:draw="urn:oasis:names:tc:opendocument:xmlns:drawing:1.0"

xmlns:presentation="urn:oasis:names:tc:opendocument:xmlns:presentation:1.0"

xmlns:dr3d="urn:oasis:names:tc:opendocument:xmlns:dr3d:1.0"

xmlns:chart="urn:oasis:names:tc:opendocument:xmlns:chart:1.0"

xmlns:form="urn:oasis:names:tc:opendocument:xmlns:form:1.0"

xmlns:script="urn:oasis:names:tc:opendocument:xmlns:script:1.0"

xmlns:style="urn:oasis:names:tc:opendocument:xmlns:style:1.0"

xmlns:number="urn:oasis:names:tc:opendocument:xmlns:datastyle:1.0"

xmlns:anim="urn:oasis:names:tc:opendocument:xmlns:animation:1.0"

xmlns:dc="http://purl.org/dc/elements/1.1/"

xmlns:xlink="http://www.w3.org/1999/xlink"

xmlns:math="http://www.w3.org/1998/Math/MathML"

xmlns:xforms="http://www.w3.org/2002/xforms"

xmlns:fo="urn:oasis:names:tc:opendocument:xmlns:xsl-fo-compatible:1.0"

xmlns:svg="urn:oasis:names:tc:opendocument:xmlns:svg-compatible:1.0"

xmlns:smil="urn:oasis:names:tc:opendocument:xmlns:smil-compatible:1.0"

>

## Document Processing and Conformance

Documents that conform to the OpenDocument specification may contain elements and attributes not specified within the OpenDocument schema. Such elements and attributes must not be part of a namespace that is defined within this specification and are called foreign elements and attributes.

Conforming applications either shall read documents that are valid against the OpenDocument schema if all foreign elements and attributes are removed before validation takes place, or shall write documents that are valid against the OpenDocument schema if all foreign elements and attributes are removed before validation takes place.

Conforming applications that read and write documents may preserve foreign elements and attributes.

In addition to this, conforming applications should preserve meta information and the content of styles. This means:

* The various <style:\*-properties> elements (see section 15) may have arbitrary attributes attached and may have arbitrary element content. All attributes attached to these elements and elements contained within these elements should be preserved (see section 15.1.3);
* elements contained within the <office:meta> element may have arbitrary element content and should be preserved (see section 2.2.1).

Foreign elements may have an office:process-content attribute attached that has the value true or false. If the attribute's value is true, or if the attribute does not exist, the element's content should be processed by conforming applications. Otherwise conforming applications should not process the element's content, but may only preserve its content. If the element's content should be processed, the document itself shall be valid against the OpenDocument schema if the unknown element is replaced with its content only.

Conforming applications shall read documents containing processing instructions and should preserve them.

There are no rules regarding the elements and attributes that actually have to be supported by conforming applications, except that applications should not use foreign elements and attributes for features defined in the OpenDocument schema. See also appendix D.

<define name="office-process-content">

<optional>

<attribute name="office:process-content" a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

## White-Space Processing and EOL Handling

In conformance with the **W3C** XML specification [XML1.0], optional white-space characters that are contained in elements that have element content (in other words that must contain elements only but not text) are ignored. This applies to the following white-space and end-of-line (**EOL**) [UNICODE] characters:

* HORIZONTAL TABULATION (0x0009)
* LINE FEED (0x000A)
* CARRIAGE RETURN (0x000D)
* SPACE (0x0020)

For any other element, white-spaces are preserved by default. Unless otherwise stated, there is no special processing for any of the four white-space characters. For some elements, different white-space processing may take place, for example the paragraph element.

The XML specification also requires that any of the four white-space characters that is contained in an attribute value is normalized to a SPACE character.

One of the following characters may be used to represent line ends:

* LINE FEED
* CARRIAGE RETURN
* The sequence of the characters CARRIAGE RETURN and LINE FEED

Conforming to the XML specification, all the possible line ends are normalized to a single LINE FEED character.

As a consequence of the white-space and EOL processing rules, any CARRIAGE RETURN characters that are contained either in the text content of an element or in an attribute value must be encoded by the character entity &#x0D;. The same applies to the HORIZONTAL TABULATION and LINE FEED characters if they are contained in an attribute value.

## MIME Types and File Name Extensions

Appendix C contains a list of MIME types and file name extensions to be used for office documents that conform to this specification and that are contained in a package (see section 2.1). This MIME types and extensions either have been registered following the procedures described in [RFC2048], or a registration is in progress.

Office documents that conform to this specification but are not contained in a package should use the MIME type text/xml.

Only MIME types and extensions that have been registered according to [RFC2048] should used for office documents that conform to this specification. The MIME types and extensions listed in appendix C should be used where appropriate.

1. Document Structure

This chapter introduces the structure of the OpenDocument format. The chapter contains the following sections:

* Document Roots
* Document Metadata
* Body Element and Document Types
* Application Settings
* Scripts
* Font Face Declarations
* Styles
* Page Styles and Layout

In the OpenDocument format, each structural component is represented by an **element**, with associated **attributes**. The structure of a document in OpenDocument format applies to all document types. There is no difference between a text document, a spreadsheet or a drawing, apart from the content. Also, all document types may contain different styles. Document content that is common to all document types can be exchanged from one type of document to another.

## Document Roots

A **document root element** is the primary element of a document in OpenDocument format. It contains the entire document. All types of documents, for example, text documents, spreadsheets, and drawing documents use the same types of document root elements.

The OpenDocument format supports the following two ways of document representation:

* As a single XML document.
* As a collection of several subdocuments within a package (see section 17), each of which stores part of the complete document. Each subdocument has a different document root and stores a particular aspect of the XML document. For example, one subdocument contains the style information and another subdocument contains the content of the document. All types of documents, for example, text and spreadsheet documents, use the same document and subdocuments definitions.

There are four types of subdocuments, each with different root elements. Additionally, the single XML document has its own root element, for a total of five different supported root elements. The root elements are summarized in the following table:

| Root Element | Subdocument Content | Subdoc. Name in Package |
| --- | --- | --- |
| <office:document> | Complete office document in a single XML document. | n/a |
| <office:document-content> | Document content and automatic styles used in the content. | content.xml |
| <office:document-styles> | Styles used in the document content and automatic styles used in the styles themselves. | styles.xml |
| <office:document-meta> | Document meta information, such as the author or the time of the last save action. | meta.xml |
| <office:document-settings> | Application-specific settings, such as the window size or printer information. | settings.xml |

The definitions of the root elements described in the table above are analogous to the definition of <office:document>, except that the child element specification is suitably restricted.

<start>

<choice>

<ref name="office-document"/>

<ref name="office-document-content"/>

<ref name="office-document-styles"/>

<ref name="office-document-meta"/>

<ref name="office-document-settings"/>

</choice>

</start>

### Document Root Element Content Models

The content models of the five root elements is summarized in the following table. Note that <office:document> may contain all supported top-level elements. None of the four subdocument root elements contain the complete data, but four combined do.

| Root Element | meta­data | app. sett. | script | font decls | style | auto style | mast style | body |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| <office:document> |  |  |  |  |  |  |  |  |
| <office:document-content> |  |  |  |  |  |  |  |  |
| <office:document-styles> |  |  |  |  |  |  |  |  |
| <office:document-meta> |  |  |  |  |  |  |  |  |
| <office:document-settings> |  |  |  |  |  |  |  |  |

The <office:document> root contains a complete document:

<define name="office-document">

<element name="office:document">

<ref name="office-document-attrs"/>

<ref name="office-document-common-attrs"/>

<ref name="office-meta"/>

<ref name="office-settings"/>

<ref name="office-scripts"/>

<ref name="office-font-face-decls"/>

<ref name="office-styles"/>

<ref name="office-automatic-styles"/>

<ref name="office-master-styles"/>

<ref name="office-body"/>

</element>

</define>

The <office:document-content> root contains only the document content, along with the automatic styles needed for the document content:

<define name="office-document-content">

<element name="office:document-content">

<ref name="office-document-common-attrs"/>

<ref name="office-scripts"/>

<ref name="office-font-face-decls"/>

<ref name="office-automatic-styles"/>

<ref name="office-body"/>

</element>

</define>

The <office:document-styles> root contains all named styles of a document, along with the automatic styles needed for the named styles:

<define name="office-document-styles">

<element name="office:document-styles">

<ref name="office-document-common-attrs"/>

<ref name="office-font-face-decls"/>

<ref name="office-styles"/>

<ref name="office-automatic-styles"/>

<ref name="office-master-styles"/>

</element>

</define>

The <office:document-meta> root contains the meta information about a document.

<define name="office-document-meta">

<element name="office:document-meta">

<ref name="office-document-common-attrs"/>

<ref name="office-meta"/>

</element>

</define>

The <office:document-settings> root contains application specific settings to be applied when processing this document.

<define name="office-document-settings">

<element name="office:document-settings">

<ref name="office-document-common-attrs"/>

<ref name="office-settings"/>

</element>

</define>

### Document Root Attributes

#### Version

All root elements take an office:version attribute, which indicates which version of this specification it complies with. The version number is in the format revision.version. If the file has a version known to an XML processor, it may validate the document. Otherwise, it is optional to validate the document, but the document must be well formed.

<define name="office-document-common-attrs" combine="interleave">

<optional>

<attribute name="office:version">

<ref name="string"/>

</attribute>

</optional>

</define>

#### MIME Type

The <office:document> element takes an office:mimetype attribute, which indicates the type of document (text, spreadsheet etc.). This attribute is especially important for flat XML files, where this is the only way the type of document can be detected (in a package, the MIME type is also present in a separate file, see section 17.4). Its values are the MIME types that are used for the packaged variant of office documents (see section 1.7).

<define name="office-document-attrs" combine="interleave">

<attribute name="office:mimetype">

<ref name="string"/>

</attribute>

</define>

## Document Metadata

Metadata is general information about a document. In the OpenDocument format, all of the metadata elements are contained in an <office:meta> element, usually located at start of the document. Metadata elements may be omitted or occur multiple times. It is application-specific how to update multiple instances of the same elements.

<define name="office-meta">

<optional>

<element name="office:meta">

<ref name="office-meta-content"/>

</element>

</optional>

</define>

<define name="office-meta-content">

<ref name="anyElements"/>

</define>

<define name="office-meta-content-strict">

<zeroOrMore>

<ref name="office-meta-data"/>

</zeroOrMore>

</define>

### Pre-Defined vs. Custom Metadata

In the OpenDocument schema the metadata is comprised of pre-defined metadata elements, user defined metadata, as well as custom metadata elements. The pre-defined metadata elements have defined semantics. They should be processed and updated by editing applications. They can be referenced from within the document through the use of suitable text fields.

User-defined metadata is a more generic mechanism which specifies a triplet of name, type, and value. Supporting applications can present these value to the user, making use of the supplied data type. The user-defined metadata can be referenced from within the document through the use of suitable text fields.

Custom metadata are arbitrary elements inside <office:meta>. Since their semantics is not defined in this specification, conforming applications in general cannot process or display this data. Applications should preserve this data when editing the document.

### Sample Metadata

Example: Sample metadata of a document in **OpenDocument** format

<office:meta>

<dc:title>Title of the document</dc:title>

<dc:description>Description/Comment for the document</dc:description>

<meta:initial-creator>User Name</meta:initial-creator>

<meta:creation-date>1999-10-18T12:34:56</meta:creation-date>

<dc:creator>User Name</dc:creator>

<dc:date>1999-10-19T15:16:17</dc:date>

<meta:printed-by>User Name</meta:printed-by>

<meta:print-date>1999-10-20T16:17:18</meta:print-date>

<dc:subject>Description of the document</dc:subject>

<meta:editing-duration>PT5H10M10S</meta:editing-duration>

<meta:keyword>First keyword</meta:keyword>

<meta:keyword>Second keyword</meta:keyword>

<meta:keyword>Third keyword</meta:keyword>

<meta:template xlink:type="simple"

xlink:href="file:///c|/office52/share/template/german/finance/budget.vor"

xlink:title="Template name"

meta:date="1999-10-15T10:11:12" />

<meta:auto-reload

xlink:type="simple"

xlink:href="file:///..."

meta:delay="P60S" />

<dc:language>de-DE</dc:language>

<meta:user-defined meta:name="Field 1"

meta:value-type="string">Value 1</meta:user-defined>

<meta:user-defined meta:name="Field 2"

meta:value-type="float">1.234</meta:user-defined>

</office:meta>

## Body Element and Document Types

The document body contains an element to indicate which type of content this document contains. Currently supported document types are:

* text documents
* drawing documents
* presentation documents
* spreadsheet documents
* chart documents
* image documents

All document types share the same content elements, but different document types place different restrictions on which elements may occur, and in what combinations. The document content is typically framed by a prelude and epilogue, which contain additional information for a specific type of document, like form data or variable declarations.

<define name="office-body">

<element name="office:body">

<ref name="office-body-content"/>

</element>

</define>

### Text Documents

The content of text documents mainly consists of a sequence containing any number of paragraphs, tables, indices, text frames, text sections, and graphical elements. Additionally, a text document may contain forms, change tracking information and variable declarations. Each of these is defined in the document prelude, and may be referenced from the document content.

<define name="office-body-content" combine="choice">

<element name="office:text">

<ref name="office-text-attlist"/>

<ref name="office-text-content-prelude"/>

<zeroOrMore>

<ref name="office-text-content-main"/>

</zeroOrMore>

<ref name="office-text-content-epilogue"/>

</element>

</define>

#### Text Document Content Model

The text document prelude contains the document's form data, change tracking information, and variable declarations. To allow office applications to implement functionality that usually is available in spreadsheets for text documents, it may also contain elements that implement enhanced table features. See also section 2.3.4.

<define name="office-text-content-prelude">

<ref name="office-forms"/>

<ref name="text-tracked-changes"/>

<ref name="text-decls"/>

<ref name="table-decls"/>

</define>

The main document content contains any sequence of text content elements, which includes paragraphs (and headings), text sections (and indices), tables, and graphical shapes. As an alternative, a text document may contain of a single page sequence.

It is not required that a text document contains a paragraph. A text document may consist of a sequence frames only.

<define name="office-text-content-main">

<choice>

<zeroOrMore>

<ref name="text-content"/>

</zeroOrMore>

<group>

<ref name="text-page-sequence"/>

<zeroOrMore>

<choice>

<ref name="draw-a"/>

<ref name="shape"/>

</choice>

</zeroOrMore>

</group>

</choice>

</define>

<define name="text-content">

<choice>

<ref name="text-h"/>

<ref name="text-p"/>

<ref name="text-list"/>

<ref name="text-numbered-paragraph"/>

<ref name="table-table"/>

<ref name="draw-a"/>

<ref name="text-section"/>

<ref name="text-soft-page-break"/>

<ref name="text-table-of-content"/>

<ref name="text-illustration-index"/>

<ref name="text-table-index"/>

<ref name="text-object-index"/>

<ref name="text-user-index"/>

<ref name="text-alphabetical-index"/>

<ref name="text-bibliography"/>

<ref name="shape"/>

<ref name="change-marks"/>

</choice>

</define>

There are no text documents specific epilogue elements, but the epilogue may contain elements that implement enhanced table features. See also section 2.3.4.

<define name="office-text-content-epilogue">

<ref name="table-functions"/>

</define>

#### Global Text Documents

There is a common use case for large documents to be edited in separate entities, such that there is a 'global' document, containing several linked constituent subdocuments. This can be implemented by using linked text sections (see section 4.4). To facilitate an editing application adapting the user interface to better support the notion of 'global' document with constituent parts (as opposed to a document with arbitrary linked content), the text:global flag can be used. If set to true, it informs applications that linked sections in this document have part-of semantics. The actual XML representation of the sections does not change.

<define name="office-text-attlist" combine="interleave">

<optional>

<attribute name="text:global" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Use Soft Page Breaks

The text:use-soft-page-breaks attribute specifies whether the document contains soft page breaks.

A soft page break is a page break that a has been included by a page oriented processor at a position where the document itself does not include a page break (e.g. by using the fo:break-before and fo:break-after formatting properties described in section 15.5.2).

Soft page breaks are specified by the <text:soft-page-break> elements described in sections 4.7 and 5.1.1:Soft Page breaks.

The use of the <text:soft-page-break> elements is always optional. An application generating the format may include the element if it has computed a paginated layout. A consuming application may handle the element while computing the layout, but it shall not depend on its existence. Soft page breaks are only supported within text documents.

A generating application that stores soft page breaks shall indicate this by setting the text:use-page-breaks attribute to true. A generating application that does not store soft page breaks shall indicate that by omitting this attribute, or by setting it to false.

An application that does not support pagination and soft page-breaks, that modifies an OpenDocument file, which includes soft page-breaks, shall at least set the text:use-page-breaks attribute to false (or remove it). It should also remove the text:soft-page-break elements from the document but is not required to do so.

An application that computes a paginated layout of a document should provide a facility to turn on export of soft page breaks for the purposes of consistent page breaks and for proper conversion to digital talking book formats (such as [DAISY]).

For <text:soft-page-break> elements that appear within table rows, the maximum number of <text:soft-page-break> elements that appear within the single table cells determines the number of page breaks that appear within the table row. The <text:soft-page-break> elements contained in each cell determine the positions where these page breaks appear within the cell content.

Similarly, <text:soft-page-break> elements that appear within text boxes and other content displayed outside the text flow, do not start a new page, but only indicate where the text-box's content breaks between two pages.

<define name="office-text-attlist" combine="interleave">

<optional>

<attribute name="text:use-soft-page-breaks" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

### Drawing Documents

The content of drawing document consists of a sequence of draw pages.

<define name="office-body-content" combine="choice">

<element name="office:drawing">

<ref name="office-drawing-attlist"/>

<ref name="office-drawing-content-prelude"/>

<ref name="office-drawing-content-main"/>

<ref name="office-drawing-content-epilogue"/>

</element>

</define>

<define name="office-drawing-attlist">

<empty/>

</define>

#### Drawing Document Content Model

The drawing document prelude may contain text declarations only. To allow office applications to implement functionality that usually is available in spreadsheets for drawing documents, it may also contain elements that implement enhanced table features. See also section 2.3.4.

<define name="office-drawing-content-prelude">

<ref name="text-decls"/>

<ref name="table-decls"/>

</define>

The main document content contains a sequence of draw pages.

<define name="office-drawing-content-main">

<zeroOrMore>

<ref name="draw-page"/>

</zeroOrMore>

</define>

There are no drawing documents specific epilogue elements, but the epilogue may contain elements that implement enhanced table features. See also section 2.3.4.

<define name="office-drawing-content-epilogue">

<ref name="table-functions"/>

</define>

### Presentation Documents

The content of presentation document consists of a sequence of draw pages.

<define name="office-body-content" combine="choice">

<element name="office:presentation">

<ref name="office-presentation-attlist"/>

<ref name="office-presentation-content-prelude"/>

<ref name="office-presentation-content-main"/>

<ref name="office-presentation-content-epilogue"/>

</element>

</define>

<define name="office-presentation-attlist">

<empty/>

</define>

#### Presentation Document Content Model

The presentation document prelude equals the one of a drawing document, but may contain some additional declarations. See also section 2.3.2.

<define name="office-presentation-content-prelude">

<ref name="text-decls"/>

<ref name="table-decls"/>

<ref name="presentation-decls"/>

</define>

The main document content contains a sequence of draw pages.

<define name="office-presentation-content-main">

<zeroOrMore>

<ref name="draw-page"/>

</zeroOrMore>

</define>

The epilogue of presentation documents may contain presentation settings. Additionally, it may contain elements that implement enhanced table features. See also section 2.3.4.

<define name="office-presentation-content-epilogue">

<ref name="presentation-settings"/>

<ref name="table-functions"/>

</define>

### Spreadsheet Documents

The content of spreadsheet documents mainly consists of a sequence of tables. Additionally, a spreadsheet document may contain forms, change tracking information and various kinds of declarations that simplify the usage of spreadsheet tables and their analysis. Each of these are contained in either the document prelude, or the document epilogue.

<define name="office-body-content" combine="choice">

<element name="office:spreadsheet">

<ref name="office-spreadsheet-attlist"/>

<ref name="office-spreadsheet-content-prelude"/>

<ref name="office-spreadsheet-content-main"/>

<ref name="office-spreadsheet-content-epilogue"/>

</element>

</define>

#### Spreadsheet Document Content Model

The spreadsheet document prelude contains the document's form data, change tracking information, calculation setting for formulas, validation rules for cell content and declarations for label ranges.

<define name="office-spreadsheet-content-prelude">

<optional>

<ref name="table-tracked-changes"/>

</optional>

<ref name="text-decls"/>

<ref name="table-decls"/>

</define>

<define name="table-decls">

<optional>

<ref name="table-calculation-settings"/>

</optional>

<optional>

<ref name="table-content-validations"/>

</optional>

<optional>

<ref name="table-label-ranges"/>

</optional>

</define>

The main document is a list of tables.

<define name="office-spreadsheet-content-main">

<zeroOrMore>

<ref name="table-table"/>

</zeroOrMore>

</define>

The epilogue of spreadsheet documents contains declarations for named expressions, database ranges, data pilot tables, consolidation operations and DDE links.

<define name="office-spreadsheet-content-epilogue">

<ref name="table-functions"/>

</define>

<define name="table-functions">

<optional>

<ref name="table-named-expressions"/>

</optional>

<optional>

<ref name="table-database-ranges"/>

</optional>

<optional>

<ref name="table-data-pilot-tables"/>

</optional>

<optional>

<ref name="table-consolidation"/>

</optional>

<optional>

<ref name="table-dde-links"/>

</optional>

</define>

### Chart Documents

The content of chart documents mainly consists of a chart element.

<define name="office-body-content" combine="choice">

<element name="office:chart">

<ref name="office-chart-attlist"/>

<ref name="office-chart-content-prelude"/>

<ref name="office-chart-content-main"/>

<ref name="office-chart-content-epilogue"/>

</element>

</define>

<define name="office-chart-attlist">

<empty/>

</define>

#### Chart Document Content Model

To allow office applications to implement functionality that usually is available in spreadsheets for the table that may be contained in a chart, the chart document prelude may contain elements that implement enhanced table features. See also section 2.3.4.

<define name="office-chart-content-prelude">

<ref name="text-decls"/>

<ref name="table-decls"/>

</define>

The main document is a chart element only.

<define name="office-chart-content-main">

<ref name="chart-chart"/>

</define>

There are no chart documents specific epilogue elements, but the epilogue may contain elements that implement enhanced table features. See also section 2.3.4.

<define name="office-chart-content-epilogue">

<ref name="table-functions"/>

</define>

### Image Documents

The content of an image document is a frame element only. The frame element must contain a single image element.

<define name="office-body-content" combine="choice">

<element name="office:image">

<ref name="office-image-attlist"/>

<ref name="office-image-content-prelude"/>

<ref name="office-image-content-main"/>

<ref name="office-image-content-epilogue"/>

</element>

</define>

<define name="office-image-attlist">

<empty/>

</define>

#### Image Document Content Model

The image document prelude is empty.

<define name="office-image-content-prelude">

<empty/>

</define>

The main document content contains a frame only.

<define name="office-image-content-main">

<ref name="draw-frame"/>

</define>

There are no image documents specific epilogue elements.

<define name="office-image-content-epilogue">

<empty/>

</define>

## Application Settings

Application settings are contained in a <office:settings> element.

<define name="office-settings">

<optional>

<element name="office:settings">

<oneOrMore>

<ref name="config-config-item-set"/>

</oneOrMore>

</element>

</optional>

</define>

The settings for office applications may be divided into several categories each represented by a <config:config-item-set> element. For instance the following two categories may exist:

* Document settings, for example default printer.
* View settings, for example zoom level.

### Sequence of Settings

The <config:config-item-set> element is a container element for all types of setting elements. The settings can be contained in the element is any order.

<define name="config-config-item-set">

<element name="config:config-item-set">

<ref name="config-config-item-set-attlist"/>

<ref name="config-items"/>

</element>

</define>

<define name="config-items">

<oneOrMore>

<choice>

<ref name="config-config-item"/>

<ref name="config-config-item-set"/>

<ref name="config-config-item-map-named"/>

<ref name="config-config-item-map-indexed"/>

</choice>

</oneOrMore>

</define>

#### Config Name

The config:name attribute identifies the name of the setting container. For top level <config:config-item-set> elements, that are elements that are direct children of the <office:settings> element, the name should be preceded by a namespace prefix that identifies the application the settings belong to.

<define name="config-config-item-set-attlist" combine="interleave">

<attribute name="config:name">

<ref name="string"/>

</attribute>

</define>

Example:

<office:settings>

<config:config-item-set xmlns:ooo="http://www.openoffice.org/...";

config:name="ooo:view-settings">

<config:config-item config:name="ViewAreaTop"

config:type="int">0</config:config-item>

</config:config-item-set>

</office:settings>

### Base Settings

The <config:config-item> element contains all base settings. The value of the setting is stored in the element.

<define name="config-config-item">

<element name="config:config-item">

<ref name="config-config-item-attlist"/>

<text/>

</element>

</define>

#### Config Name

The config:name attribute identifies the name of the setting.

<define name="config-config-item-attlist" combine="interleave">

<attribute name="config:name">

<ref name="string"/>

</attribute>

</define>

#### Config Type

The config:type attribute identifies the data type of setting.

<define name="config-config-item-attlist" combine="interleave">

<attribute name="config:type">

<choice>

<value>boolean</value>

<value>short</value>

<value>int</value>

<value>long</value>

<value>double</value>

<value>string</value>

<value>datetime</value>

<value>base64Binary</value>

</choice>

</attribute>

</define>

### Index Access of Sequences

The <config:config-item-map-indexed> element is a container element for sequences. The order specifies the index of the elements

<define name="config-config-item-map-indexed">

<element name="config:config-item-map-indexed">

<ref name="config-config-item-map-indexed-attlist"/>

<oneOrMore>

<ref name="config-config-item-map-entry"/>

</oneOrMore>

</element>

</define>

#### Config Name

The config:name attribute identifies the name of the setting sequence.

<define name="config-config-item-map-indexed-attlist" combine="interleave">

<attribute name="config:name">

<ref name="string"/>

</attribute>

</define>

### Map Entry

The <config:config-item-map-entry> element represents an entry in an indexed or named settings sequence. It is a container element for all types of setting elements.

<define name="config-config-item-map-entry">

<element name="config:config-item-map-entry">

<ref name="config-config-item-map-entry-attlist"/>

<ref name="config-items"/>

</element>

</define>

#### Config Name

The config:name attribute identifies the name of the setting sequence.

<define name="config-config-item-map-entry-attlist" combine="interleave">

<optional>

<attribute name="config:name">

<ref name="string"/>

</attribute>

</optional>

</define>

### Name Access of Sequences

The <config:config-item-map-named> element is a container element for sequences, where each setting in the sequence is identified by its name.

<define name="config-config-item-map-named">

<element name="config:config-item-map-named">

<ref name="config-config-item-map-named-attlist"/>

<oneOrMore>

<ref name="config-config-item-map-entry"/>

</oneOrMore>

</element>

</define>

#### Config Name

The config:name attribute identifies the name of the setting sequence.

<define name="config-config-item-map-named-attlist" combine="interleave">

<attribute name="config:name">

<ref name="string"/>

</attribute>

</define>

### Cursor Position Setting

A common view setting for editing applications is the position where the text cursor was while saving the document. For WYSIWYG applications, this usually will be a position within a paragraph only. For applications that provide an XML based view of the document, the cursor position could be also between arbitrary elements, or even within tags.

To represent a text cursor position within a document, a processing instruction with PITarget opendocument (see §2.6 of [XML1.0]) should be used. The name of the cursor position processing instruction, cursor-position, shall follow the PITarget opendocument. The processing instruction may have arbitrary application specific attributes, for instance to connect the cursor position with a certain view of the document, where the views themselves are specified as application specific settings. The syntax for these attributes shall be the same as for attributes within XML start tags.

Where a text cursor position is not sufficient to recreate a document view, applications may use arbitrary document specific settings in addition to the cursor position processing instruction. They may also use arbitrary document specific settings if the cursor position is not a text cursor position, but for instance a selection of drawing objects.

Example: cursor position processing instruction

<text:p>This is<?opendocument cursor-position view-id="view1"?> an example.</text:p>

## Scripts

A document may contain several scripts in different scripting languages. Each script is represented by a <office:script> element. All these script elements are contained in a single <office:scripts> element.

Scripts do not imply a scripting language or an object model. A script can for instance operate on the Document Object Model (DOM) composed from the XML representation of a document in OpenDocument format (see [DOM2]), or on an application specific API.

Scripts cannot modify a document while the document is loading. However, some events are called immediately after the document is loaded.

In addition to <office:script> elements, the <office:scripts> element may also contain an <office:event-listeners> element which contains the events assigned to the document itself. Examples for these are events called when the document is opened or closed. See section 12.4 for more information on the <office:event-listeners> element.

<define name="office-scripts">

<optional>

<element name="office:scripts">

<zeroOrMore>

<ref name="office-script"/>

</zeroOrMore>

<optional>

<ref name="office-event-listeners"/>

</optional>

</element>

</optional>

</define>

### Script

The <office:script> element contains script language specific content. In most situations, the element contains the source code of the script, but it may also contain a compiled version of the script or a link to some external script code.

<define name="office-script">

<element name="office:script">

<ref name="office-script-attlist"/>

<mixed>

<ref name="anyElements"/>

</mixed>

</element>

</define>

#### Script Language

The attribute script:language specifies the language of the script by its name. Since script language names are application specific, the name should be preceded by a namespace prefix.

<define name="office-script-attlist">

<attribute name="script:language">

<ref name="string"/>

</attribute>

</define>

## Font Face Declarations

A document in OpenDocument format may contain font face declarations. A font face declaration provides information about the fonts used by the author of a document, so that these fonts or fonts that are very close to these fonts may be located on other systems. See section 14.6 for details.

<define name="office-font-face-decls">

<optional>

<element name="office:font-face-decls">

<zeroOrMore>

<ref name="style-font-face"/>

</zeroOrMore>

</element>

</optional>

</define>

## Styles

The OpenDocument format supports the following types of styles:

* **Common styles**  
  Most office applications support styles within their user interface. Within this specification, the XML representations of such styles are referred to as styles. When a differentiation from the other types of styles is required, they are referred to as common styles. The term *common* indicates that this is the type of style that an office application user considers to be a style.
* **Automatic styles**  
  An automatic style contains formatting properties that, in the user interface view of a document, are assigned to an object such as a paragraph. The term *automatic* indicates that the style is generated automatically. In other words, formatting properties that are immediately assigned to a specific object are represented by an automatic style. This way, a separation of content and layout is achieved.
* **Master styles**  
  A master style is a common style that contains formatting information and additional content that is displayed with the document content when the style is applied. An example of a master style are master pages. Master pages can be used in graphical applications. In this case, the additional content is any drawing shapes that are displayed as the background of the draw page. Master pages can also be used in text documents. In this case, the additional content is the headers and footers. Please note that the content that is contained within master styles is additional content that influences the representation of a document but does not change the content of a document.

As far as the office application user is concerned, all types of styles are part of the document. They represent the output device-independent layout and formatting information that the author of a document has used to create or edit the document. The assumption is that the author of the document wants this formatting and layout information to be preserved when the document is reloaded or displayed on any device, because this is common practice for documents created by word processors.

This type of style information differs from [CSS2] or [XSLT] style sheets that are used to display a document. An additional style sheet for CSS, XSLT, and so on, is required to display a document in OpenDocument format on a certain device. This style sheet must take into account the styles in the document as well as the requirements and capabilities of the output device. The ideal case is that this style sheet depends on the output device only.

See section 14 for more information on styles.

### Location of Styles

Common and automatic styles have the same XML representation, but they are contained within two distinct container elements, as follows:

* <office:styles> for common styles
* <office:automatic-styles> for automatic styles
* Master styles are contained within a container element of its own:
* <office:master-styles>

<define name="office-styles">

<optional>

<element name="office:styles">

<interleave>

<ref name="styles"/>

<zeroOrMore>

<ref name="style-default-style"/>

</zeroOrMore>

<optional>

<ref name="text-outline-style"/>

</optional>

<zeroOrMore>

<ref name="text-notes-configuration"/>

</zeroOrMore>

<optional>

<ref name="text-bibliography-configuration"/>

</optional>

<optional>

<ref name="text-linenumbering-configuration"/>

</optional>

<zeroOrMore>

<ref name="draw-gradient"/>

</zeroOrMore>

<zeroOrMore>

<ref name="svg-linearGradient"/>

</zeroOrMore>

<zeroOrMore>

<ref name="svg-radialGradient"/>

</zeroOrMore>

<zeroOrMore>

<ref name="draw-hatch"/>

</zeroOrMore>

<zeroOrMore>

<ref name="draw-fill-image"/>

</zeroOrMore>

<zeroOrMore>

<ref name="draw-marker"/>

</zeroOrMore>

<zeroOrMore>

<ref name="draw-stroke-dash"/>

</zeroOrMore>

<zeroOrMore>

<ref name="draw-opacity"/>

</zeroOrMore>

<zeroOrMore>

<ref name="style-presentation-page-layout"/>

</zeroOrMore>

</interleave>

</element>

</optional>

</define>

<define name="office-automatic-styles">

<optional>

<element name="office:automatic-styles">

<interleave>

<ref name="styles"/>

<zeroOrMore>

<ref name="style-page-layout"/>

</zeroOrMore>

</interleave>

</element>

</optional>

</define>

<define name="office-master-styles">

<optional>

<element name="office:master-styles">

<interleave>

<zeroOrMore>

<ref name="style-master-page"/>

</zeroOrMore>

<optional>

<ref name="style-handout-master"/>

</optional>

<optional>

<ref name="draw-layer-set"/>

</optional>

</interleave>

</element>

</optional>

</define>

<define name="styles">

<interleave>

<zeroOrMore>

<ref name="style-style"/>

</zeroOrMore>

<zeroOrMore>

<ref name="text-list-style"/>

</zeroOrMore>

<zeroOrMore>

<ref name="number-number-style"/>

</zeroOrMore>

<zeroOrMore>

<ref name="number-currency-style"/>

</zeroOrMore>

<zeroOrMore>

<ref name="number-percentage-style"/>

</zeroOrMore>

<zeroOrMore>

<ref name="number-date-style"/>

</zeroOrMore>

<zeroOrMore>

<ref name="number-time-style"/>

</zeroOrMore>

<zeroOrMore>

<ref name="number-boolean-style"/>

</zeroOrMore>

<zeroOrMore>

<ref name="number-text-style"/>

</zeroOrMore>

</interleave>

</define>

The following examples illustrate the different types of OpenDocument styles.

Example: OpenDocument styles

<office:document ...>

<office:styles>

...

</office:styles>

<office:automatic-styles>

...

</office:automatic-styles>

<office:master-styles>

...

</office:master-styles>

</office:document>

## Page Styles and Layout

The style and layout of the pages in a document is determined by:

* Page Layouts
* Master Pages

A **page layout** describes the physical properties or geometry of a page, for example, page size, margins, header height, and footer height.

A **master page** is a template for pages in a document. It contains a reference to a page layout which specifies the physical properties of the page and can also contain static content that is displayed on all pages in the document that use the master page. Examples of static content are headers, footers, or background graphics.

If a text or spreadsheet document is displayed in a paged layout, the master pages are instantiated to generate a sequence of pages containing the document content. When a master page is instantiated, an empty page is generated with the properties of the page master and the static content of the master page. The body of the page is then filled with content. If multiple pages in a document use the same master page, the master page can be instantiated several times within the document.

In text and spreadsheet documents, a master page can be assigned to paragraph and table styles using a style:master-page-name attribute. Each time the paragraph or table style is applied to text, a page break is inserted before the paragraph or table. The page that starts at the page break position uses the specified master page.

In drawings and presentations, master pages can be assigned to drawing pages using a style:parent-style-name attribute.

**Note:** The OpenDocument paging methodology differs significantly from the methodology used in [XSL]. In XSL, headers and footers are contained within page sequences that also contain the document content. In the OpenDocument format, headers and footers are contained in page styles. With either approach, the content of headers and footers can be changed or omitted without affecting the document content.

Page layouts are described in section 14.3. Master pages are described in section 14.4.

1. Metadata Elements

The metadata elements borrow heavily upon the metadata standards developed by the Dublin Core Metadata Initiative ([http://www.dublincore.org](http://www.dublincore.org/)). Metadata elements drawn directly from the Dublin Core work use its namespace prefix (see section 1.3).

## Pre-Defined Metadata Elements

There is a set of pre-defined metadata elements which should be processed and updated by the applications. Metadata elements may be omitted or occur multiple times. It is application-specific how to update multiple instances of the same elements.

### Generator

The <meta:generator> element contains a string that identifies the application or tool that was used to create or last modify the XML document. This string should match the definition for user-agents in the HTTP protocol a specified in section 14.43 of [RFC2616]. The generator string should allow product versions to differ between all released versions of a user agent, for instance by including build ids or patch level information.

Conforming applications may use the generator string to work around bugs that exist or existed in certain applications, but shall not deliberately implement a different behavior depending on a certain generator string.

If the application that created the document could not provide an identifier string, the application does not export this element. If another application modifies the document and it cannot provide a unique identifier, it shall not export the original identifier belonging to the application that created the document.

<define name="office-meta-data" combine="choice">

<element name="meta:generator">

<ref name="string"/>

</element>

</define>

### Title

The <dc:title> element specifies the title of the document.

<define name="office-meta-data" combine="choice">

<element name="dc:title">

<ref name="string"/>

</element>

</define>

### Description

The <dc:description> element contains a brief description of the document.

<define name="office-meta-data" combine="choice">

<element name="dc:description">

<ref name="string"/>

</element>

</define>

### Subject

The <dc:subject> element specifies the subject of the document.

<define name="office-meta-data" combine="choice">

<element name="dc:subject">

<ref name="string"/>

</element>

</define>

### Keywords

The <meta:keyword> element contains a pertaining to the document. The metadata can contain any number of <meta:keyword> elements, each element specifying one keyword.

<define name="office-meta-data" combine="choice">

<element name="meta:keyword">

<ref name="string"/>

</element>

</define>

### Initial Creator

The <meta:initial-creator> element specifies the name of the person who created the document initially.

<define name="office-meta-data" combine="choice">

<element name="meta:initial-creator">

<ref name="string"/>

</element>

</define>

### Creator

The <dc:creator> element specifies the name of the person who last modified the document. The name of this element was chosen for compatibility with the Dublin Core, but this definition of "creator" used here differs from Dublin Core, which defines creator as "An entity primarily responsible for making the content of the resource." In OpenDocument terminology, the last person to modify the document is primarily responsible for making the content of the document.

<define name="office-meta-data" combine="choice">

<ref name="dc-creator"/>

</define>

<define name="dc-creator">

<element name="dc:creator">

<ref name="string"/>

</element>

</define>

### Printed By

The <meta:printed-by> element specifies the name of the last person who printed the document.

<define name="office-meta-data" combine="choice">

<element name="meta:printed-by">

<ref name="string"/>

</element>

</define>

### Creation Date and Time

The <meta:creation-date> element specifies the date and time when the document was created initially.

To conform with [xmlschema-2], the date and time format is YYYY-MM-DDThh:mm:ss.

<define name="office-meta-data" combine="choice">

<element name="meta:creation-date">

<ref name="dateTime"/>

</element>

</define>

### Modification Date and Time

The <dc:date> element specifies the date and time when the document was last modified.

To conform with [xmlschema-2], the date and time format is YYYY-MM-DDThh:mm:ss.

The name of this element was chosen for compatibility with the Dublin Core.

<define name="office-meta-data" combine="choice">

<ref name="dc-date"/>

</define>

<define name="dc-date">

<element name="dc:date">

<ref name="dateTime"/>

</element>

</define>

### Print Date and Time

The <meta:print-date> element specifies the date and time when the document was last printed.

To conform with [xmlschema-2], the date and time format is YYYY-MM-DDThh:mm:ss.

<define name="office-meta-data" combine="choice">

<element name="meta:print-date">

<ref name="dateTime"/>

</element>

</define>

### Document Template

The <meta:template> element contains a URL for the document template that was used to create the document. The URL is specified as an XLink.

This element conforms to the XLink Specification. See [XLink].

The attributes that may be associated with the <meta:template> element are:

* Template location
* Template title
* Template modification date and time

#### Template Location

An xlink:href attribute specifies the location of the document template.

#### Template Title

The xlink:title attribute specifies the name of the document template.

#### Template Modification Date and Time

The meta:date attribute specifies the date and time when the template was last modified, prior to being used to create the current document.

To conform with [xmlschema-2], the date and time format is YYYY-MM-DDThh:mm:ss.

<define name="office-meta-data" combine="choice">

<element name="meta:template">

<attribute name="xlink:href">

<ref name="anyURI"/>

</attribute>

<optional>

<attribute name="xlink:type" a:defaultValue="simple">

<value>simple</value>

</attribute>

</optional>

<optional>

<attribute name="xlink:actuate" a:defaultValue="onRequest">

<value>onRequest</value>

</attribute>

</optional>

<optional>

<attribute name="xlink:title">

<ref name="string"/>

</attribute>

</optional>

<optional>

<attribute name="meta:date">

<ref name="dateTime"/>

</attribute>

</optional>

</element>

</define>

### Automatic Reload

The <meta:auto-reload> element specifies whether a document is reloaded or replaced by another document after a certain period of time has elapsed.

The attributes that may be associated with the <meta:auto-reload> element are:

* Reload URL
* Reload delay

#### Reload URL

If a loaded document should be replaced by another document after a certain period of time, the <meta:auto-reload> element is presented as an XLink. An xlink:href attribute identifies the URL of the replacement document.

#### Reload Delay

The meta:delay attribute specifies the reload delay.

To conform with the duration data type of [xmlschema-2], the format of the value of this attribute is PnYnMnDTnHnMnS. See §3.2.6 of [xmlschema-2] for more detailed information on this duration format.

<define name="office-meta-data" combine="choice">

<element name="meta:auto-reload">

<optional>

<attribute name="xlink:type" a:defaultValue="simple">

<value>simple</value>

</attribute>

</optional>

<optional>

<attribute name="xlink:show" a:defaultValue="replace">

<value>replace</value>

</attribute>

</optional>

<optional>

<attribute name="xlink:actuate" a:defaultValue="onLoad">

<value>onLoad</value>

</attribute>

</optional>

<optional>

<attribute name="xlink:href">

<ref name="anyURI"/>

</attribute>

</optional>

<optional>

<attribute name="meta:delay">

<ref name="duration"/>

</attribute>

</optional>

</element>

</define>

### Hyperlink Behavior

The <meta:hyperlink-behaviour> element specifies the default behavior for hyperlinks in the document.

The only attribute that may be associated with the <meta:hyperlink-behaviour> element is:

* Target frame

#### Target Frame

The meta:target-frame-name attribute specifies the name of the default target frame in which to display a document referenced by a hyperlink.

This attribute can have one of the following values:

* \_self : The referenced document replaces the content of the current frame.
* \_blank : The referenced document is displayed in a new frame.
* \_parent : The referenced document is displayed in the parent frame of the current frame.
* \_top : The referenced document is displayed in the topmost frame, that is the frame that contains the current frame as a child or descendent but is not contained within another frame.
* A frame name : The referenced document is displayed in the named frame. If the named frame does not exist, a new frame with that name is created.

To conform with the XLink Specification, an additional xlink:show attribute is attached to the <meta:hyperlink-behaviour> element. If the value of the meta:target-frame-name attribute is \_blank, the xlink:show attribute value is new. If the value of the meta:target-frame-name attribute is any of the other value options, the value of the xlink:show attribute is replace.

<define name="office-meta-data" combine="choice">

<element name="meta:hyperlink-behaviour">

<optional>

<attribute name="office:target-frame-name">

<ref name="targetFrameName"/>

</attribute>

</optional>

<optional>

<attribute name="xlink:show">

<choice>

<value>new</value>

<value>replace</value>

</choice>

</attribute>

</optional>

</element>

</define>

### Language

The <dc:language> element specifies the default language of the document.

The manner in which the language is represented is similar to the language tag described in [RFC3066]. It consists of a two or three letter Language Code taken from the ISO 639 standard optionally followed by a hyphen (-) and a two-letter Country Code taken from the ISO 3166 standard.

<define name="office-meta-data" combine="choice">

<element name="dc:language">

<ref name="language"/>

</element>

</define>

### Editing Cycles

The <meta:editing-cycles> element specifies the number of editing cycles the document has been through.

The value of this element is incremented every time the document is saved. The element contains the number of editing cycles as text.

<define name="office-meta-data" combine="choice">

<element name="meta:editing-cycles">

<ref name="nonNegativeInteger"/>

</element>

</define>

### Editing Duration

The <meta:editing-duration> element specifies the total time spent editing the document.

The duration is represented in the duration data type of [xmlschema-2], that is PnYnMnDTnHnMnS. See §3.2.6 of [xmlschema-2] for more detailed information on this duration format.

<define name="office-meta-data" combine="choice">

<element name="meta:editing-duration">

<ref name="duration"/>

</element>

</define>

### Document Statistics

The <meta:document-statistic> element specifies the statistics of the document, for example, the page count, word count, and so on. The statistics are specified as attributes of the <meta:document-statistic> element and the statistics that are exported with the document depend on the document type and the application used to create the document.

| Document Type | Document Statistics Attributes |
| --- | --- |
| Text | meta:page-count meta:table-count meta:draw-count meta:image-count meta:object-count meta:ole-object-count meta:paragraph-count  meta:word-count meta:character-count meta:row-count meta:frame-count meta:sentence-count meta:syllable-count meta:non-whitespace-character-count |
| Spreadsheet | meta:page-count meta:table-count meta:image-count meta:cell-count meta:object-count |
| Graphic | meta:page-count meta:image-count meta:object-count |

<define name="office-meta-data" combine="choice">

<element name="meta:document-statistic">

<optional>

<attribute name="meta:page-count">

<ref name="nonNegativeInteger"/>

</attribute>

</optional>

<optional>

<attribute name="meta:table-count">

<ref name="nonNegativeInteger"/>

</attribute>

</optional>

<optional>

<attribute name="meta:draw-count">

<ref name="nonNegativeInteger"/>

</attribute>

</optional>

<optional>

<attribute name="meta:image-count">

<ref name="nonNegativeInteger"/>

</attribute>

</optional>

<optional>

<attribute name="meta:ole-object-count">

<ref name="nonNegativeInteger"/>

</attribute>

</optional>

<optional>

<attribute name="meta:object-count">

<ref name="nonNegativeInteger"/>

</attribute>

</optional>

<optional>

<attribute name="meta:paragraph-count">

<ref name="nonNegativeInteger"/>

</attribute>

</optional>

<optional>

<attribute name="meta:word-count">

<ref name="nonNegativeInteger"/>

</attribute>

</optional>

<optional>

<attribute name="meta:character-count">

<ref name="nonNegativeInteger"/>

</attribute>

</optional>

<optional>

<attribute name="frame-count">

<ref name="nonNegativeInteger"/>

</attribute>

</optional>

<optional>

<attribute name="sentence-count">

<ref name="nonNegativeInteger"/>

</attribute>

</optional>

<optional>

<attribute name="syllable-count">

<ref name="nonNegativeInteger"/>

</attribute>

</optional>

<optional>

<attribute name="non-whitespace-character-count">

<ref name="nonNegativeInteger"/>

</attribute>

</optional>

<optional>

<attribute name="meta:row-count">

<ref name="nonNegativeInteger"/>

</attribute>

</optional>

<optional>

<attribute name="meta:cell-count">

<ref name="nonNegativeInteger"/>

</attribute>

</optional>

</element>

</define>

## User-defined Metadata

The <meta:user-defined> element specifies any additional user-defined metadata for the document. Each instance of this element can contain one piece of user-defined metadata. The element contains:

* A meta:name attribute, which identifies the name of the metadata element.
* An optional meta:value-type attribute, which identifies the type of the metadata element. The allowed meta types are float, date, time, boolean and string (see also section 6.7.1).
* The value of the element, which is the metadata in the format described in section 6.7.1 as value of the office:value attributes for the various data types.

The default type for meta-data elements is string.

<define name="office-meta-data" combine="choice">

<element name="meta:user-defined">

<attribute name="meta:name">

<ref name="string"/>

</attribute>

<choice>

<group>

<attribute name="meta:value-type">

<value>float</value>

</attribute>

<ref name="double"/>

</group>

<group>

<attribute name="meta:value-type">

<value>date</value>

</attribute>

<ref name="dateOrDateTime"/>

</group>

<group>

<attribute name="meta:value-type">

<value>time</value>

</attribute>

<ref name="duration"/>

</group>

<group>

<attribute name="meta:value-type">

<value>boolean</value>

</attribute>

<ref name="boolean"/>

</group>

<group>

<attribute name="meta:value-type">

<value>string</value>

</attribute>

<ref name="string"/>

</group>

<text/>

</choice>

</element>

</define>

## Custom Metadata

In addition to the pre-defined metadata elements, applications should also preserve any additional content found inside the <office:meta> element. As there is no semantics specified for such foreign content, applications need not process this information other than to preserve it when editing the document.

1. Text Content

## Headings, Paragraphs and Basic Text Structure

This section describes the XML elements and attributes that are used to represent heading and paragraph components in a text document.

The elements <text:h> and <text:p> represent headings and paragraphs, respectively, and are collectively referred to as paragraph elements. All text content in an OpenDocument file must be contained in either of these elements.

### Headings

Headings define the chapter structure for a document. A chapter or subchapter begins with a heading and extends to the next heading at the same or higher level.

<define name="text-h">

<element name="text:h">

<ref name="heading-attrs"/>

<ref name="paragraph-attrs"/>

<optional>

<ref name="text-number"/>

</optional>

<zeroOrMore>

<ref name="paragraph-content"/>

</zeroOrMore>

</element>

</define>

#### Heading Level

The text:outline-level attribute associated with the heading element determines the level of the heading, starting with 1. Headings without a level attribute are assumed to be at level 1.

<define name="heading-attrs" combine="interleave">

<attribute name="text:outline-level">

<ref name="positiveInteger"/>

</attribute>

</define>

#### Heading Numbering

Header numbering can be changed by additional attributes, similar to those on list items (see section 4.3.2, below). The numbering of headers can be restarted by setting the text:restart-numbering attribute to true.

<define name="heading-attrs" combine="interleave">

<optional>

<attribute name="text:restart-numbering" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Start Value

The attribute text:start-value may be used to restart the numbering of headers of the current header's level, by setting a new value for the numbering.

<define name="heading-attrs" combine="interleave">

<optional>

<attribute name="text:start-value">

<ref name="nonNegativeInteger"/>

</attribute>

</optional>

</define>

#### Suppress Header Numbering

It is sometimes desired to have a specific heading which should not be numbered. This corresponds to unnumbered list headers in lists (see sections 4.3). To facilitate this, an optional attribute text:is-list-header can be used. If true, the given header will not be numbered, even if an explicit list-style is given.

<define name="heading-attrs" combine="interleave">

<optional>

<attribute name="text:is-list-header" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Formatted Heading Number

If a heading has a numbering applied, the text of the formatted number can be included in a <text:number> element. This text can be used by applications that do not support numbering of headings, but it will be ignored by applications that support numbering.

<define name="text-number">

<element name="text:number">

<ref name="string"/>

</element>

</define>

### Paragraphs

Paragraphs are the basic unit of text.

<define name="text-p">

<element name="text:p">

<ref name="paragraph-attrs"/>

<zeroOrMore>

<ref name="paragraph-content"/>

</zeroOrMore>

</element>

</define>

### Common Paragraph Elements Attributes

The paragraph elements have text:style-name, text:class-names and text:cond-style-name attributes. These attributes must reference paragraph styles.

A text:style-name attribute references a paragraph style, while a text:cond-style-name attribute references a conditional-style, that is, a style that contains conditions and maps to other styles (see section 14.1.1). If a conditional style is applied to a paragraph, the text:style-name attribute contains the name of the style that was the result of the conditional style evaluation, while the conditional style name itself is the value of the text:cond-style-name attribute. This XML structure simplifies [XSLT] transformations because XSLT only has to acknowledge the conditional style if the formatting attributes are relevant. The referenced style can be a common style or an automatic style.

A text:class-names attribute takes a whitespace separated list of paragraph style names. The referenced styles are applied in the order they are contained in the list. If both, text:style-name and text:class-names are present, the style referenced by the text:style-name attribute is as the first style in the list in text:class-names. If a conditional style is specified together with a style:class-names attribute, but without the text:style-name attribute, then the first style in the style list is used as the value of the missing text:style-name attribute.

Conforming applications should support the text:class-names attribute and also should preserve it while editing.

<define name="paragraph-attrs">

<optional>

<attribute name="text:style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

<optional>

<attribute name="text:class-names">

<ref name="styleNameRefs"/>

</attribute>

</optional>

<optional>

<attribute name="text:cond-style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

</define>

Example: Styles and conditional styles

<text:p text:style-name="Heading 1">

"Heading 1" is not a conditional style.

</text:p>

<text:p text:style-name="Numbering 1" text:cond-style-name="Text body">

"Text body" is a conditional style. If it is contained in a numbered

paragraph, it maps to "Numbering 1". This is assumed in this example.

</text:p>

A paragraph may have an ID. This ID can be used to reference the paragraph from other elements.

<define name="paragraph-attrs" combine="interleave">

<optional>

<ref name="text-id"/>

</optional>

</define>

## Page Sequences

A page sequence element <text:page-sequence> specifies a sequence of master pages that are instantiated in exactly the same order as they are referenced in the page sequence. If a text document contains a page sequence, it will consist of exactly as many pages as specified. Documents with page sequences do not have a main text flow consisting of headings and paragraphs as is the case for documents that do not contain a page sequence. Text content is included within text boxes for documents with page sequences. The only other content that is permitted are drawing objects.

Example: Page Sequence

<style:automatic-style>

<style:page-layout name="pm1">

<!-- portrait page -->

</style:page-layout>

<style:page-layout name="pm2">

<!-- landscape page -->

</style:page-layout>

</style:automatic-style>

...

<style:master-styles>

<style:master-page name="portrait" style:page-layout-name="pm1"/>

<style:master-page name="landscape" style:page-layout-name="pm2"/>

</style:master-styles>

...

<office:body>

<text:page-sequence>

<text:page text:master-page-name="portrait"/>

<text:page text:master-page-name="portrait"/>

<text:page text:master-page-name="landscape"/>

<text:page text:master-page-name="landscape"/>

<text:page text:master-page-name="portrait"/>

</text:page-sequence>

<draw:frame ...>

<draw:text-box ...>

<text:p>Example text.</text:p>

...

</draw:text-box>

</draw:frame>

</office:body>

<define name="text-page-sequence">

<element name="text:page-sequence">

<oneOrMore>

<ref name="text-page"/>

</oneOrMore>

</element>

</define>

### Page

The <text:page> element specifies a single page within a page sequence.

<define name="text-page">

<element name="text:page">

<ref name="text-page-attlist"/>

<empty/>

</element>

</define>

#### Master Page Name

The text:master-page-name attribute specifies the master page that is instantiated.

<define name="text-page-attlist">

<attribute name="text:master-page-name">

<ref name="styleNameRef"/>

</attribute>

</define>

## Lists

The OpenDocument format supports list structures, similar to those found in [HTML4]. A list is a paragraph-level element, which contains an optional list header, followed by a sequence of list items. The list header and each list item contains a sequence of paragraph or list elements. Lists can be nested.

Lists may be numbered. The numbering may be restarted with a specific numbering at each list item. Lists may also continue numbering from other lists, allowing the user to merge several lists into a single, discontinuous list. Note that whether the list numbering is displayed depends on a suitable list style being used.

In addition to this structural information, lists can have list styles associated with them, which contain the relevant layout information, such as

* the type of list item label, such as bullet or number,
* list item label width and distance,
* bullet character or image (if any),
* number format for the bullet numbering (if any),
* paragraph indent for list items.

### List Block

A list is represented by the <text:list> element. It contains an optional list header, followed by any number of list items.

Every list has a list level, which is determined by the nesting of the <text:list> elements. If a list is not contained within another list, the list level is 1. If the list in contained within another list, the list level is the list level of the list in which is it contained incremented by one. If a list is contained in a table cell or text box, the list level returns to 1, even though the table or textbox itself may be nested within another list.

The attributes that may be associated with the list element are:

* Style name
* Continue numbering

<define name="text-list">

<element name="text:list">

<ref name="text-list-attr"/>

<optional>

<ref name="text-list-header"/>

</optional>

<zeroOrMore>

<ref name="text-list-item"/>

</zeroOrMore>

</element>

</define>

#### Style Name

The optional text:style-name attribute specifies the name of the list style that is applied to the list.

If this attribute is not included and therefore no list style is specified, one of the following actions is taken:

* If the list is contained within another list, the list style defaults to the style of the surrounding list.
* If there is no list style specified for the surrounding list, but the list contains paragraphs that have paragraph styles attached specifying a list style, this list style is used for any of these paragraphs.
* A default list style is applied to any other paragraphs.

To determine which formatting properties are applied to a list, the list level and list style name are taken into account. See section 14.10 for more information on list formatting properties.

<define name="text-list-attr" combine="interleave">

<optional>

<attribute name="text:style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

</define>

#### Continue Numbering

By default, the first list item in a list starts with the number specified in the list style. The continue numbering attribute can be used to continue the numbering from the preceding list.

This attribute can be used with the <text:list> element and can have a value of true or false.

If the value of the attribute is true and the numbering style of the preceding list is the same as the current list, the number of the first list item in the current list is the number of the last item in the preceding list incremented by one.

<define name="text-list-attr" combine="interleave">

<optional>

<attribute name="text:continue-numbering">

<ref name="boolean"/>

</attribute>

</optional>

</define>

### List Item

List items contain the textual content of a list. A <text:list-item> element can contain paragraphs, headings, lists or soft page breaks. A list item cannot contain tables.

<define name="text-list-item">

<element name="text:list-item">

<ref name="text-list-item-attr"/>

<ref name="text-list-item-content"/>

</element>

</define>

<define name="text-list-item-content">

<optional>

<ref name="text-number"/>

</optional>

<zeroOrMore>

<choice>

<ref name="text-p"/>

<ref name="text-h"/>

<ref name="text-list"/>

<ref name="text-soft-page-break"/>

</choice>

</zeroOrMore>

</define>

The first line in a list item is preceded by a bullet or number, depending on the list style assigned to the list. If a list item starts another list immediately and does not contain any text, no bullet or number is displayed.

The only attribute that may be associated with the <text:list-item> element is:

* Start value

#### Start Value

The numbering of the current list can be restarted at a certain number. The text:start-value attribute is used to specify the number with which to restart the list.

This attribute can only be applied to items in a list with a numbering list style. It restarts the numbering of the list at the current item.

<define name="text-list-item-attr" combine="interleave">

<optional>

<attribute name="text:start-value">

<ref name="nonNegativeInteger"/>

</attribute>

</optional>

</define>

#### Formatted Number

If a list item has a numbering applied, the text of the formatted number can be included in a <text:number> element. This text can be used by applications that do not support numbering, but it will be ignored by applications that support numbering. See also section 4.1.1.

Example: Lists and sublists

<text:list text:style-name="List 1">

<text:list-item>

<text:p>This is the first list item</text:p>

<text:p>This is a continuation of the first list item.</text:p>

</text:list-item>

<text:list-item>

<text:p>This is the second list item.

It contains a sub list.</text:p>

<text:list>

<text:list-item><text:p>This is a sub list item.</text:p>

</text:list-item>

<text:list-item><text:p>This is a sub list item.</text:p>

</text:list-item>

<text:list-item><text:p>This is a sub list item.</text:p>

</text:list-item>

</text:list>

</text:list-item>

<text:list-item>

<text:p>This is the third list item</text:p>

</text:list-item>

</text:list>

### List Header

A list header is a special kind of list item. It contains one or more paragraphs that are displayed before a list. The paragraphs are formatted like list items but they do not have a preceding number or bullet. The list header is represented by the list header element.

<define name="text-list-header">

<element name="text:list-header">

<ref name="text-list-item-content"/>

</element>

</define>

### Numbered Paragraphs

In some instances, it is desirable to specify a list not as a structural element comprising of several list items, but to determine on a per-paragraph level whether the paragraph is numbered, and at which level. To facilitate this, the <text:numbered-paragraph> element allows the numbering of an individual paragraph, as if it was part of a list at a specified level.

Numbered paragraphs may use the same continuous numbering properties that list items use, and thus form an equivalent, alternative way of specifying lists. A list in <text:list> representation could be converted into a list in <text:numbered-paragraph> representation and vice versa.

<define name="text-numbered-paragraph">

<element name="text:numbered-paragraph">

<ref name="text-numbered-paragraph-attr"/>

<optional>

<ref name="text-number"/>

</optional>

<choice>

<ref name="text-p"/>

<ref name="text-h"/>

</choice>

</element>

</define>

A numbered paragraph can be assigned a list level. A numbered paragraph is equivalent to a list nested to the given level, containing one list item with one paragraph. If no level is given, the numbered paragraph is interpreted as being on level 1.

<define name="text-numbered-paragraph-attr" combine="interleave">

<optional>

<attribute name="text:level" a:defaultValue="1">

<ref name="positiveInteger"/>

</attribute>

</optional>

</define>

As a numbered paragraph combines the functionality of a (possibly nested) list with a single list item, it can also use the attributes of those elements.

<define name="text-numbered-paragraph-attr" combine="interleave">

<ref name="text-list-attr"/>

</define>

<define name="text-numbered-paragraph-attr" combine="interleave">

<ref name="text-list-item-attr"/>

</define>

The text of a formatted number can be included in a <text:number> element. This text can be used by applications that do not support numbering, but it will be ignored by applications that support numbering. See also section 4.1.1.

## Text Sections

A text section is a named region of paragraph-level text content. Sections start and end on paragraph boundaries and can contain any number of paragraphs.

Sections have two uses in the OpenDocument format: They can be used to assign certain formatting properties to a region of text. They can also be used to group text that is automatically acquired from some external data source.

In addition to Sections can contain regular text content or the text can be contained in another file and linked to the section. Sections can also be write-protected or hidden.

Sections can have settings for text columns, background color or pattern, and notes configuration. These settings form the section style, which is represented in a <style:style> element. See section 14.8.3 for details.

The formatting properties for sections are explained in section 15.7.

Sections support two ways of linking to external content. If a section is linked to another document, the link can be through one of the following:

* A resource identified by an XLink, represented by a text:section-source element
* Dynamic Data Exchange (DDE), represented by a office:dde-source element

Linking information for external content is contained in the section element's first child. A section that links to external content contains the full representation of the data source, so that processors need to understand the linking information only if they wish to update the contents of the section.

<define name="text-section">

<element name="text:section">

<ref name="text-section-attr"/>

<choice>

<ref name="text-section-source"/>

<ref name="text-section-source-dde"/>

<empty/>

</choice>

<zeroOrMore>

<ref name="text-content"/>

</zeroOrMore>

</element>

</define>

**Note:** List items may not contain sections. Thus, lists may only be wholly contained within section elements. If it is desired to achieve the effect of overlapping lists and sections, or of sections contained within lists, the lists must be split into several lists, each of which would then be wholly contained within a section. When splitting the list, suitable attributes for continuous numbering should be set such that display and behavior are the same as with the original list not interrupted by sections.

### Section Attributes

Text indices, described in chapter 7, may be considered a special kind of text section, as they share the same general structure as well as certain attributes. These are combined in the following definition:

<define name="text-section-attr" combine="interleave">

<ref name="sectionAttr"/>

</define>

The remaining attributes in this section are specific to the <text:section> element.

#### Section Style

The text:style-name attribute refers to a section style.

<define name="sectionAttr" combine="interleave">

<optional>

<attribute name="text:style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

</define>

#### Section Name

Every section must have a name that uniquely identifies the section. The text:name attribute contains the name of the section.

<define name="sectionAttr" combine="interleave">

<attribute name="text:name">

<ref name="string"/>

</attribute>

</define>

#### Protected Sections

A section can be protected, which means that a user can not edit the section. The text:protected attribute indicates whether or not a section is protected. The user interface must enforce the protection attribute if it is enabled.

<define name="sectionAttr" combine="interleave">

<optional>

<attribute name="text:protected">

<ref name="boolean"/>

</attribute>

</optional>

</define>

A user can use the user interface to reset the protection flag, unless the section is further protected by a password. In this case, the user must know the password in order to reset the protection flag. The text:protection-key attribute specifies the password that protects the section. To avoid saving the password directly into the XML file, only a hash value of the password is stored.

<define name="sectionAttr" combine="interleave">

<optional>

<attribute name="text:protection-key">

<ref name="string"/>

</attribute>

</optional>

</define>

#### Hidden Sections and Conditional Sections

Sections can be hidden based on a condition or they can be hidden unconditionally.

The text:display attribute specifies whether or not the section is hidden. The value of this attribute can be:

* true, the section is displayed. This is the default setting.
* none, the section is hidden unconditionally.
* condition, the section is hidden under the condition specified in the text:condition attribute.

The text:condition attribute specifies the condition under which the section is hidden. The condition is encoded as a string. If the value of text:display is condition, the text:condition attribute must be present.

<define name="text-section-attr" combine="interleave">

<choice>

<attribute name="text:display">

<choice>

<value>true</value>

<value>none</value>

</choice>

</attribute>

<group>

<attribute name="text:display">

<value>condition</value>

</attribute>

<attribute name="text:condition">

<ref name="string"/>

</attribute>

</group>

<empty/>

</choice>

</define>

### Section Source

The <text:section-source> element indicates that the enclosed section is a linked section. If this element is used, it must be the first element in the <text:section> element.

<define name="text-section-source">

<element name="text:section-source">

<ref name="text-section-source-attr"/>

</element>

</define>

The attributes that may be associated with the <text:section-source> attribute are:

* Section source URL
* Name of linked section
* Filter name

#### Section Source URL

These attributes identify the document or section to which the section is linked. The name of the target section is identified by the local part of the URL, following the hash mark. The xlink:href attribute is implied because <text:section-source> elements may also link to internal sections.

<define name="text-section-source-attr" combine="interleave">

<optional>

<attribute name="xlink:href">

<ref name="anyURI"/>

</attribute>

<optional>

<attribute name="xlink:type" a:defaultValue="simple">

<value>simple</value>

</attribute>

</optional>

<optional>

<attribute name="xlink:show" a:defaultValue="embed">

<value>embed</value>

</attribute>

</optional>

</optional>

</define>

#### Name of Linked Section

If the link targets a section of a document, the attribute text:section name contains the name of the target section. If the attribute is not present, the link targets the entire document.

<define name="text-section-source-attr" combine="interleave">

<optional>

<attribute name="text:section-name">

<ref name="string"/>

</attribute>

</optional>

</define>

#### Filter Name

The text:filter-name attribute specifies which filter type was used to import the link target. The value of this attribute is implementation dependent.

<define name="text-section-source-attr" combine="interleave">

<optional>

<attribute name="text:filter-name">

<ref name="string"/>

</attribute>

</optional>

</define>

### DDE Source

If sections are linked via DDE, their linking information is represented by <office:dde-source> elements. It contains attributes that specify the application, topic and item of the DDE connection. Note that because the section contains the XML rendition of the DDE link's content, this information only needs to be processed if updated data from the DDE link are desired.

See section 12.6 for the use of DDE connections.

<define name="text-section-source-dde">

<ref name="office-dde-source"/>

</define>

## Page-bound graphical content

Within text documents, images, embedded objects and other drawing objects appear at the level of a paragraph if they are anchored to a page rather than to a paragraph or a character position within a paragraph. See section 9.2 for details on drawing objects, and section 9.2.16 for their anchoring.

## Change Tracking

This section describes how changes in text documents can be represented.

### Tracked Changes

All tracked changes to text documents are stored in a list. The list contains an element for each change made to the document. If the <text:tracked-changes> element is absent, change tracking is not enabled.

<define name="text-tracked-changes">

<optional>

<element name="text:tracked-changes">

<ref name="text-tracked-changes-attr"/>

<zeroOrMore>

<ref name="text-changed-region"/>

</zeroOrMore>

</element>

</optional>

</define>

#### Track Changes

This attribute determines whether or not user agents should track and record changes for this document.

<define name="text-tracked-changes-attr" combine="interleave">

<optional>

<attribute name="text:track-changes" a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

### Changed Regions

For every changed region of a document, there is one entry in the list of tracked changes. This entry contains a list of all changes that were applied to the region. The start and end of this region are marked by the start and end elements that are described in the next section.

<define name="text-changed-region">

<element name="text:changed-region">

<ref name="text-changed-region-attr"/>

<ref name="text-changed-region-content"/>

</element>

</define>

#### Change ID

Every element has an ID. The elements that mark the start and end of a region use this ID to identify the region to which they belong.

<define name="text-changed-region-attr" combine="interleave">

<attribute name="text:id">

<ref name="ID"/>

</attribute>

</define>

### Insertion

The <text:insertion> element contains the information that is required to identify any insertion of content. This content can be a piece of text within a paragraph, a whole paragraph, or a whole table. The inserted content is part of the text document itself and is marked by a change start and a change end element.

<define name="text-changed-region-content" combine="choice">

<element name="text:insertion">

<ref name="office-change-info"/>

</element>

</define>

**Example**: Insertion of text

<text:tracked-changes>

<text:changed-region text:id="c001">

<text:insertion>

<office:change-info>

<dc:creator>Michael Brauer</dc:creator>

<dc:date>1999-05-18T12:56:04</dc:date>

</office:change-info>

</text:insertion>

</text:changed-region>

</text:tracked-changes>

<text:p>

This is the original text<text:change-start text:change-id="c001"/>,

but this has been added<text:change-end text:change-id="c001"/>.

</text:p>

### Deletion

A <text:deletion> element contains content that was deleted while change tracking was enabled. The position where the text was deleted is marked by the change position element.

If part of a paragraph was deleted, the text that was deleted is contained in this element as a paragraph element. If the deleted text is reinserted into the document, the paragraph is joined with the paragraph where the deletion took place.

<define name="text-changed-region-content" combine="choice">

<element name="text:deletion">

<ref name="office-change-info"/>

<zeroOrMore>

<ref name="text-content"/>

</zeroOrMore>

</element>

</define>

**Example:** Deletion of text

<text:tracked-changes>

<text:changed-region text:id="c002">

<text:deletion>

<office:change-info>

<dc:creator>Michael Brauer</dc:creator>

<dc:date>1999-05-18T12:56:04</dc:date>

</office:change-info>

<text:p>, but this has been deleted</text:p>

</text:deletion>

</text:changed-region>

</text:tracked-changes>

<text:p>

This is the original text<text:change text:region-id="c002"/>.

</text:p>

This example shows:

* Deleted text = , but this has been deleted   
  This text is contained in the <text:p> element within the <text:deletion> element.
* Current text = This is the original text.   
  This text is contained in the <text:p> element at the end of the example.
* Original text before deletion took place = This is the original text, but this has been deleted.

Note that the deleted text, like all text in the OpenDocument format, is contained in a paragraph element. To reconstruct the original text, this paragraph is merged with its surrounding. In other words, a deletion consisting of only a single word would be represented as a paragraph containing the word.

To reconstruct the text before the deletion took place, do:

* If the change mark is inside a paragraph, insert the text content of the <text:deletion> element as if the beginning <text:p> and final </text:p> tags were missing.
* If the change mark is inside a header, proceed as above, except adapt the end tags to match their new counterparts.
* Otherwise, simply copy the text content of the <text:deletion> element in place of the change mark.

**Example**: Given the following change:

<text:changed-region text:id="example">

<text:deletion>

<office:change-info>...</office:change-info>

<text:p>Hello</text:p>

<text:p>World!</text:p>

</text:deletion>

</text:changed-region>

The first (and most common) case occurs if a change mark is inside a regular paragraph:

<text:p>abc<text:change text:id="example/>def</text:p>

To reconstruct the original text, the two <text:p> elements are copied to replace the change mark, except the beginning and ending tags are missing:

<text:p>abcHello</text:p>

<text:p>World!def</text:p>

If the change mark occurred inside a header, the same procedure is followed, except the copied tags are adapted to make sure we still have well-formed XML.

<text:h>abc<text:change text:id="example/>def</text:h>

becomes:

<text:h>abcHello</text:h>

<text:h>World!def</text:h>

The third case occurs when a change occurs outside of a paragraph. In this case, the deleted text is simply copied verbatim.

<text:p>abcdef</text:p>

<text:change text:id="example/>

<text:p>ghijkl</text:p>

This becomes:

<text:p>abcdef</text:p>

<text:p>Hello</text:p>

<text:p>World!</text:p>

<text:p>ghijkl</text:p>

If, in the first two cases, the deletion contains complete paragraphs, then additional empty paragraphs must be put into the <text:deletion> element to achieve the desired result.

The change that took place from

<text:p>abc</text:p>

<text:h>Hello</text:h>

<text:h>World!</text:h>

<text:p>def</text:p>

to

<text:p>abc<text:change text:id="example/>def</text:p>

would be represented as:

<text:changed-region text:id="example">

<text:deletion>

<office:change-info>...</office:change-info>

<text:p/>

<text:h>Hello</text:h>

<text:h>World!</text:h>

<text:p/>

</text:deletion>

</text:changed-region>

### Format Change

A format change element represents any change in formatting attributes. The region where the change took place is marked by a change start and a change end element.

<define name="text-changed-region-content" combine="choice">

<element name="text:format-change">

<ref name="office-change-info"/>

</element>

</define>

**Note:** A format change element does not contain the actual changes that took place.

### Change Info

The change info element contains meta information who made the change and when. It is also used for spreadsheet documents, and thus described in a section 12.3 (Change Tracking Metadata).

### Change Marks

There are three elements that mark the start and the end of a changed region, as follows:

* Change start element – <text:change-start>   
  This element marks the start of a region with content where text has been inserted or the format has been changed.
* Change end element – <text:change-end>  
  This element marks the end of a region with content where text has been inserted or the format has been changed.
* Change position element – <text:change>  
  This element marks a position in an empty region where text has been deleted.

All three elements have an attribute that specifies the ID of the region to which they belong.

<define name="change-marks">

<choice>

<element name="text:change">

<ref name="change-mark-attr"/>

</element>

<element name="text:change-start">

<ref name="change-mark-attr"/>

</element>

<element name="text:change-end">

<ref name="change-mark-attr"/>

</element>

</choice>

</define>

<define name="change-mark-attr">

<attribute name="text:change-id">

<ref name="IDREF"/>

</attribute>

</define>

## Soft Page Break

The <text:soft-page-break> element represents a soft page break.

See section 2.3.1:Use Soft Page BreaksUse Soft Page Breaks for details regarding soft page breaks.

<define name="text-soft-page-break">

<element name="text:soft-page-break">

<empty/>

</element>

</define>

## Text Declarations

Several text elements need per-document declarations before they can be used. For example, variable fields require that the variables used are being declared at the beginning of the document. These declarations are collected at the beginning of a text document. All such declarations are optional. The detailed description for each declaration can be found in the appropriate chapter.

The supported text declarations are:

* variable declarations – These declarations are used for variable fields. (cf. section 6.3.1).
* user field declarations – These declarations are used for user-defined fields (cf. section 6.3.5).
* sequence declarations – These declarations are used for sequence fields (cf. section 6.3.8).
* DDE connections – These declarations are used for DDE fields and DDE sections (cf. sections 6.6.9 and 4.4.3, respectively).
* auto mark file – This declaration is used for generation of alphabetical indices (cf. section 7.8.2).

<define name="text-decls">

<optional>

<element name="text:variable-decls">

<zeroOrMore>

<ref name="text-variable-decl"/>

</zeroOrMore>

</element>

</optional>

<optional>

<element name="text:sequence-decls">

<zeroOrMore>

<ref name="text-sequence-decl"/>

</zeroOrMore>

</element>

</optional>

<optional>

<element name="text:user-field-decls">

<zeroOrMore>

<ref name="text-user-field-decl"/>

</zeroOrMore>

</element>

</optional>

<optional>

<element name="text:dde-connection-decls">

<zeroOrMore>

<ref name="text-dde-connection-decl"/>

</zeroOrMore>

</element>

</optional>

<optional>

<ref name="text-alphabetical-index-auto-mark-file"/>

</optional>

</define>

1. Paragraph Elements Content

## Basic Text Content

Paragraph element's children make up the text content of any document. All text contained in a paragraph element or their children is text content, with few exceptions detailed later. This should significantly ease transformations into other formats, since transformations may ignore any child elements of paragraph elements and only process their text content, and still obtain a faithful representation of text content.

Text content elements that do not contain in-line text children are:

* (foot- and end-)notes (see section 5.3)

Foot- and endnotes contain text content, but are typically displayed outside the main text content, e.g., at the end of a page or document.

* rubies (see section 5.4)

Ruby texts are usually displayed above or below the main text.

* annotations (see section 5.5)

Annotations are typically not displayed.

<define name="paragraph-content" combine="choice">

<text/>

</define>

### White-space Characters

If the paragraph element or any of its child elements contains white-space characters, they are collapsed. Leading white-space characters at the paragraph start as well as trailing white-space characters at the paragraph end are ignored. In detail, the following conversions take place:

The following [UNICODE] characters are normalized to a SPACE character:

* HORIZONTAL TABULATION (0x0009)
* CARRIAGE RETURN (0x000D)
* LINE FEED (0x000A)
* SPACE (0x0020)

In addition, these characters are ignored if the preceding character is a white-space character. The preceding character can be contained in the same element, in the parent element, or in the preceding sibling element, as long as it is contained within the same paragraph element and the element in which it is contained processes white-space characters as described above. White-space characters at the start or end of the paragraph are ignored, regardless whether they are contained in the paragraph element itself, or in a child element in which white-space characters are collapsed as described above.

These white-space processing rules shall enable authors to use white-space characters to improve the readability of the XML source of an OpenDocument document in the same way as they can use them in HTML.

White-space processing takes place within the following elements:

* <text:p>
* <text:h>
* <text:span>
* <text:a>
* <text:ref-point>
* <text:ref-point-start>
* <text:ref-point-end>
* <text:bookmark>
* <text:bookmark-start>
* <text:bookmark-end>

Note: In [XSL], white-space processing of a paragraph of text can be enabled by attaching an fo:white-space="collapse" attribute to the <fo:block> element that corresponds to the paragraph element.

, in other words they areprocessed in the same way that [HTML4] processes them.

#### Space Character

In general, consecutive white-space characters in a paragraph are collapsed. For this reason, there is a special XML element used to represent the [UNICODE] character SPACE (0x0020).

This element uses an optional attribute called text:c to specify the number of SPACE characters that the element represents. A missing text:c attribute is interpreted as meaning a single SPACE character.

This element is required to represent the second and all following SPACE characters in a sequence of SPACE characters. It is not an error if the character preceding the element is not a white-space character, but it is good practice to use this element for the second and all following SPACE characters in a sequence. This way, an application recognizes a single space character without recognizing this element.

<define name="paragraph-content" combine="choice">

<element name="text:s">

<optional>

<attribute name="text:c">

<ref name="nonNegativeInteger"/>

</attribute>

</optional>

</element>

</define>

#### Tab Character

The <text:tab> element represents the [UNICODE] tab character HORIZONTAL TABULATION (0x0009) in a heading or paragraph. A <text:tab> element reserves space from the current position up to the next tab-stop, as defined in the paragraph's style information.

<define name="paragraph-content" combine="choice">

<element name="text:tab">

<ref name="text-tab-attr"/>

</element>

</define>

To determine which tab-stop a tab character will advance to requires layout information. To make it easier for non-layout oriented processors to determine this information, applications may generate a text:tab-ref attribute as a hint that associates a tab character with a tab-stop in the current paragraph style. It contains the number of the tab-stop that the tab character refers to. The position 0 has a special meaning and signifies the start margin of the paragraph.

<define name="text-tab-attr">

<optional>

<attribute name="text:tab-ref">

<ref name="nonNegativeInteger"/>

</attribute>

</optional>

</define>

**Note:** The text:tab-ref attribute is only a hint to help non-layout oriented processors to determine the tab/tab-stop association. Layout oriented processors should determine the tab positions solely based on the style information.

#### Line Breaks

The <text:line-break> element represents a line break in a heading or paragraph.

<define name="paragraph-content" combine="choice">

<element name="text:line-break">

<empty/>

</element>

</define>

#### Soft Page Break

The <text:soft-page-break> element represents a soft page break within a heading or paragraph.

See section 2.3.1:Use Soft Page BreaksUse Soft Page Breaks for details regarding soft page breaks.

<define name="paragraph-content" combine="choice">

<ref name="text-soft-page-break"/>

</define>

### Soft Hyphens, Hyphens, and Non-breaking Blanks

Soft hyphens, hyphens, and non-breaking blanks are represented by [UNICODE] characters.

|  |  |
| --- | --- |
| The [UNICODE] character... | Represents... |
| SOFT HYPHEN (00AD) | soft hyphens |
| NON-BREAKING HYPHEN (2011) | non-breaking hyphens |
| NO-BREAK SPACE (00A0) | non-breaking blanks |

### Attributed Text

The <text:span> element represents portions of text that are attributed using a certain text style or class. The content of this element is the text that uses the text style.

The name of the a text style or text class is the value of a text:style-name or text:class-names attributes, respectively, attached to the <text:span> element. These attributes must refer to text styles or classes.

A text:style-name attribute references a single text style. A text:class-names attribute takes a whitespace separated list of text style names. The referenced text styles are applied in the order they are contained in the list. If both, text:style-name and text:class-names are present, the style referenced by the text:style-name attribute is treated as the first style in the list in text:class-names. Conforming application should support the text:class-names attribute and also should preserve it while editing.

<text:span> elements can be nested.

White-space characters contained in this element are collapsed.

<define name="paragraph-content" combine="choice">

<element name="text:span">

<optional>

<attribute name="text:style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

<optional>

<attribute name="text:class-names">

<ref name="styleNameRefs"/>

</attribute>

</optional>

<zeroOrMore>

<ref name="paragraph-content"/>

</zeroOrMore>

</element>

</define>

Example: Text style in **OpenDocument** documents:

<text:p>

The last word of this sentence is

<text:span text:style-name="emphasize">emphasized</text:span>.

</text:p>

### Hyperlinks

Hyperlinks in text documents are represented by a <text:a> element.

This element also contains an event table element, <office:event-listeners>, which contains the events assigned to the hyperlink. See section 12.4 for more information on the event table element.

<define name="paragraph-content" combine="choice">

<element name="text:a">

<ref name="text-a-attlist"/>

<optional>

<ref name="office-event-listeners"/>

</optional>

<zeroOrMore>

<ref name="paragraph-content"/>

</zeroOrMore>

</element>

</define>

The attributes that may be associated with the <text:a> element are:

* Name
* Link location
* Target frame
* Text styles

#### Name

A hyperlink can have a name, but it is not essential. The office:name attribute specifies the name of the hyperlink if one exists. This name can serve as a target for some other hyperlinks.

<define name="text-a-attlist" combine="interleave">

<optional>

<attribute name="office:name">

<ref name="string"/>

</attribute>

</optional>

</define>

#### Title

The office:title attribute specifies a short accessible description for hint text.

See appendix E for guidelines how to use this attribute.

<define name="text-a-attlist" combine="interleave">

<optional>

<attribute name="office:title">

<ref name="string"/>

</attribute>

</optional>

</define>

#### Link Location

The xlink:href attribute specifies the URL for the target location of the link.

<define name="text-a-attlist" combine="interleave">

<attribute name="xlink:href">

<ref name="anyURI"/>

</attribute>

<optional>

<attribute name="xlink:type" a:defaultValue="simple">

<value>simple</value>

</attribute>

</optional>

<optional>

<attribute name="xlink:actuate" a:defaultValue="onRequest">

<value>onRequest</value>

</attribute>

</optional>

</define>

#### Target Frame

The office:target-frame-name attribute specifies the target frame of the link. This attribute can have one of the following values:

* \_self – The referenced document replaces the content of the current frame.
* \_blank – The referenced document is displayed in a new frame.
* \_parent – The referenced document is displayed in the parent frame of the current frame.
* \_top – The referenced document is displayed in the uppermost frame, that is the frame that contains the current frame as a child or descendent but is not contained within another frame.
* A frame name – The referenced document is displayed in the named frame. If the named frame does not exist, a new frame with that name is created.

To conform with the XLink Specification, an additional xlink:show attribute is attached to the <text:a> element. If the value of the attribute is \_blank, the xlink:show attribute value is new. If the value of the attribute is any of the other value options, the value of the xlink:show attribute is replace. See [XLink].

<define name="text-a-attlist" combine="interleave">

<optional>

<attribute name="office:target-frame-name">

<ref name="targetFrameName"/>

</attribute>

</optional>

<optional>

<attribute name="xlink:show">

<choice>

<value>new</value>

<value>replace</value>

</choice>

</attribute>

</optional>

</define>

#### Text Styles

Every hyperlink has two text styles as follows:

* If the link location of the hyperlink was not visited, the text style specifies by the text:style-name attribute is applied to the text of the hyperlink.
* If the link location of the hyperlink was already visited, the text style specified by the text:visited-style-name attribute is applied to the text of the hyperlink

<define name="text-a-attlist" combine="interleave">

<optional>

<attribute name="text:style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

<optional>

<attribute name="text:visited-style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

</define>

## Bookmarks and References

### Bookmarks

Bookmarks can either mark a text position or a text range. A text range can start at any text position and end at another text position. In particular, a bookmark can start in the middle of one paragraph and end in the middle of another paragraph. The XML element used to represent a bookmark varies depending on the type of bookmark, as follows:

* <text:bookmark> – to mark one text position
* <text:bookmark-start> – to mark the start position in a text range
* <text:bookmark-end> – to mark the end position in a text range

For every <text:bookmark-start> element, there must be a <text:bookmark-end> element in the same text flow using the same text:name attribute, and vice versa. The <text:bookmark-start> element must precede the <text:bookmark-end> element.

<define name="paragraph-content" combine="choice">

<choice>

<element name="text:bookmark">

<attribute name="text:name">

<ref name="string"/>

</attribute>

</element>

<element name="text:bookmark-start">

<attribute name="text:name">

<ref name="string"/>

</attribute>

</element>

<element name="text:bookmark-end">

<attribute name="text:name">

<ref name="string"/>

</attribute>

</element>

</choice>

</define>

Example: Bookmarks

<text:p>

<text:bookmark text:name="Mark 1"/>There is a text mark in front of this

paragraph.

<text:bookmark-start text:name="Mark 2"/>In front of this paragraph there is

the start of a bookmark.

</text:p>

<text:p>

This bookmark ends

<text:bookmark-end text:name="Mark 2"/>

amid this sentence.

</text:p>

### References

The representation of references is modeled on the XML representation of bookmarks. There are two types of reference marks, as follows:

* A point reference  
  A point reference marks a particular position in text and is represented by a single <text:reference-mark> element.
* A range reference  
  A range reference marks a range of characters in text and is represented by two elements; <text:reference-mark-start> to mark the start of the range and <text:reference-mark-end> to mark the end of the range.

Every reference is identified by its name, which must be unique. In a range reference, the start and end elements must use the same reference name.

#### Point References

The <text:reference-mark> element represents a point reference.

<define name="paragraph-content" combine="choice">

<element name="text:reference-mark">

<attribute name="text:name">

<ref name="string"/>

</attribute>

</element>

</define>

#### Range References

The <text:reference-mark-start> and <text:reference-mark-end> elements represent a range reference.

<define name="paragraph-content" combine="choice">

<choice>

<element name="text:reference-mark-start">

<attribute name="text:name">

<ref name="string"/>

</attribute>

</element>

<element name="text:reference-mark-end">

<attribute name="text:name">

<ref name="string"/>

</attribute>

</element>

</choice>

</define>

In the OpenDocument schema, three elements are used to represent references instead of one element because references represented as a single XML element:

* Cannot support overlapping references
* Do not interact well with other elements

Take the following example:

Example: Overlapping range references

<text:p>  
 <text:reference-mark-start text:name="first"/>This is an

<text:reference-mark-start text:name="second"/>example of a sentence

<text:reference-mark-end text:name="first"/>with overlapping references.

<text:reference-mark-end text:name="second"/>  
</text:p>

The example paragraph shows two references that cover the following text:

|  |  |
| --- | --- |
| reference “first” | “This is an example of a sentence” |
| reference “second” | “example of a sentence with overlapping references.” |

This overlapping structure cannot be represented using a single reference element to contain the referenced text. Similarly, a reference spanning multiple paragraphs creates the same situation as two overlapping XML elements, as does character formatting either starts or ends, but not both, within the referenced text.

## Notes

Notes consist of a <text:note> element which occurs in the text stream at the position to which the note is anchored. How notes are numbered and rendered is determined by <text:notes-configuration> element, which occurs inside the <office:styles> section.

### Note Element

The note element represents text notes which are attached to a certain text position. A common implementation of this concept are the footnotes and endnotes found in most word processors. A note contains a note citation element and a note body elements, which contains the note's content.

In OpenDocument documents, notes are represented in a similar fashion to footnotes in [XSL]. In XSL, the first child of the note element contains the citation in the form of an <fo:inline> element. The OpenDocument schema uses the same structure but introduces a <text:note-citation> element. The second child contains the note body, just as in XSL.

Additionally, OpenDocument features <text:notes-configuration> elements. To achieve a similar effect to the note configuration in XSL, every note citation element must be formatted appropriately.

<define name="paragraph-content" combine="choice">

<element name="text:note">

<ref name="text-note-class"/>

<optional>

<attribute name="text:id">

<ref name="string"/>

</attribute>

</optional>

<element name="text:note-citation">

<optional>

<attribute name="text:label">

<ref name="string"/>

</attribute>

</optional>

<text/>

</element>

<element name="text:note-body">

<zeroOrMore>

<ref name="text-content"/>

</zeroOrMore>

</element>

</element>

</define>

#### Note Class

Each note belongs to a class which determines how the note is expected to be rendered. Currently, two note classes are supported: Footnotes and endnotes.

<define name="text-note-class">

<attribute name="text:note-class">

<choice>

<value>footnote</value>

<value>endnote</value>

</choice>

</attribute>

</define>

#### Footnote Reference ID

The footnote reference ID is used by references to footnotes to identify the footnote that is referenced.

#### Note Citation Element

The <text:note-citation> element contains the formatted note citation element, either as a formatted number or a string.

#### Note Label

Note citation elements can be labeled or numbered. If they are numbered, the number is chosen and formatted automatically according to the notes configuration element. If they are labeled, the user must supply a label for every note he/she inserts into the document. This label is stored in the text:label attribute of the <text:note-citation> element.

#### Note Body

The <text:note-body> element contains the actual content of the footnote. It does not have any attributes.

The schema allows for the inclusion of notes into the note body. While this may be reasonable for some future note types, it is not reasonable for footnotes and endnotes. Conforming applications may or may not support such nested notes.

#### Footnote example

<text:p>

This paragraph contains a footnote

<text:note text:note-class="footnote" text:id="ftn001">

<text:note-citation>1</text:note-citation>

<text:note-body>

<text:p>

This footnote has a generated sequence number

</text:p>

</text:note-body>

</text:note>

.

</text:p>

<text:p>

This paragraph contains a footnote

<text:note text:note-class="footnote" text:id="ftn002">

<text:note-citation text:label="\*">\*</text:note-citation>

<text:note-body>

<text:p>

This footnote has a fixed citation

</text:p>

</text:note-body>

</text:note>

, too

</text:p>

## Ruby

A ruby is additional text that is displayed above or below some base text. The purpose of ruby is to annotate the base text or provide information about its pronunciation.

There are two elements that can be contained in the <text:ruby> element:

* Ruby base
* Ruby text

The <text:ruby-base> element contains the text that is to be annotated. It contains any paragraph element content, like text spans. The element's text:style-name attribute references a ruby style that specifies further formatting attributes of the ruby. See section 14.8.4 for details.

The <text:ruby-text > element contains the annotation text. It may contain only plain text. The element's text:style-name attribute references a text style that specifies further formatting attributes used for the text.

<define name="paragraph-content" combine="choice">

<element name="text:ruby">

<optional>

<attribute name="text:style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

<element name="text:ruby-base">

<ref name="paragraph-content"/>

</element>

<element name="text:ruby-text">

<optional>

<attribute name="text:style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

<text/>

</element>

</element>

</define>

## Text Annotation

The OpenDocument format allows annotation to appear within a paragraph element. See section 12.1 for details on annotations.

<define name="paragraph-content" combine="choice">

<ref name="office-annotation"/>

</define>

## Index Marks

Index marks are used to mark text areas for inclusion into text indices. They are similar in structure to bookmarks and references. They are discussed in detail section 7.1, together with text indices.

## Change Tracking and Change Marks

Paragraphs may also contain change tracking marks. These have already been explained in the chapter on change tracking (section 4.6), and are referenced here for completeness.

<define name="paragraph-content" combine="choice">

<ref name="change-marks"/>

</define>

## Inline graphics and text-boxes

Within text documents, images, embedded objects and other drawing objects may be anchored to a paragraph, to a character, or as a character. If they are anchored to a paragraph, they appear within a paragraph at an arbitrary position. If they are anchored to or as a character, they appear within a paragraph at exactly the character position they are anchored to or as. See section 9.2 for details on drawing objects, and section 9.2.16 for their anchoring.

<define name="paragraph-content" combine="choice">

<choice>

<ref name="shape"/>

<ref name="draw-a"/>

</choice>

</define>

1. Text Fields

OpenDocument text documents or OpenDocument text content embedded in other types of documents can contain variable text elements called fields. There are several different types of field, each of which implements a different type of variable text element. Fields are most commonly used for:

* Page numbers  
  A page number field displays the number of the page it appears on. This field is useful for footers. For every page on which the footer appears, the field assumes the current page number so that all pages are numbered correctly.
* Creation dates  
  A creation date field displays the date on which the current document was created. This field is useful for document templates. Every document created using the template contains the date when it was created.
* Number ranges  
  A number range field allows the user to number certain elements, for example, images or tables. A number range field displays its own position in relation to the other number range fields for the same range. Therefore, if an image and its associated number range field are moved within a document, the fields are automatically updated to reflect the new order.

This section describes how fields are represented in the OpenDocument file format.

## Common Characteristics of Field Elements

Each field type is represented by a corresponding element type. A field in a document is encoded as a single element of the appropriate type. The content of the element is the textual representation of the current field value as it would be displayed or printed. Therefore, ignoring all field elements and displaying only the textual content of the elements provides an approximate text-only version of the document.

The value of a field is usually stored in an attribute. It is necessary to store the value so that the presentation of the field can be recomputed if necessary, for example, if the user decides to change the formatting style of the field. It is also necessary to store the presentation style of the element content, to facilitate easy processing of the XML document. For example, if complete processing of a field is impossible or undesirable, the application can ignore the field and use only the content in this situation. For string values, if the value is identical to the presentation, the value attribute is omitted to avoid duplicate storage of information.

For fields that can store different types of content, for example, numbers, strings, or dates, a value type is stored in addition to the actual value. The value and value type attributes are explained later in section 6.7.1. If more information is needed to restore a field, it is stored in additional attributes.

The most common attributes of field elements are:

* Fixed fields  
  Many fields have a variant where the content does not change after the initial value is assigned. These fields are generally marked by the attribute text:fixed. See section 6.7.2 for more information on this attribute.
* Formatting style  
  Several field types, particularly those representing number, date, or time data, contain a formatting style. In the OpenDocument format, this formatting style is represented by a style:data-style-name attribute. Since the user can change the presentation style for fields, applications must be able to recompute a new representation of the field content at any time. See section 6.7.7 for more information on this attribute.

## Document Fields

OpenDocument fields can display information about the current document or about a specific part of the current document, such as the author, the current page number, or the document creation date. These fields are collectively referred to as document fields.

Document fields are often fixed. A field can be marked fixed to indicate that its content is preserved, rather than re-evaluated, when the document is edited. For example, a date field shows the current date. If the date field is marked fixed, the value of the field is preserved during subsequent edits and always reflects the original date on which the field was inserted into the document. If the field is not marked fixed, its value changes whenever the document is edited. In the same way, the author field can show the original author or the last author of a document, depending on whether the field is marked fixed or not.

The group of document fields includes:

* Date and time fields
* Page number fields
* Sender and author fields
* Chapter fields
* File name fields
* Document template fields

### Date Fields

Date fields display the current date. The date can be adjusted to display a date other than the current date. For example, the date can be changed on a document that was edited late at night so that it displays the date of the following day or several days later.

This element contains the presentation of the date field value, depending on the data style specified. The default date is the current date. The value of this element can be preserved using the text:fixed attribute described in section 6.7.2.

<define name="paragraph-content" combine="choice">

<element name="text:date">

<ref name="text-date-attlist"/>

<text/>

</element>

</define>

The attributes that may be associated with the <text:date> element are:

* Date value
* Date adjustment
* Fixed (see section 6.7.2)
* Formatting style (see section 6.7.7). The formatting style must be a date data style, see section 14.7 for more information.

<define name="text-date-attlist" combine="interleave">

<interleave>

<ref name="common-field-fixed-attlist"/>

<ref name="common-field-data-style-name-attlist"/>

</interleave>

</define>

#### Date Value

The text:date-value attribute specifies a particular date value. For example, if the date field is marked fixed, this attribute can be used to specify the date on which the field was marked as fixed. This attribute can also be used to specify a future date. Some applications support date and time in addition to date-only values.

The date value should conform with the date formats described in §3.2.7 and §3.2.9 of [xmlschema-2]. If no value is specified, the current date is assumed, even if the field is marked fixed.

<define name="text-date-attlist" combine="interleave">

<optional>

<attribute name="text:date-value">

<ref name="dateOrDateTime"/>

</attribute>

</optional>

</define>

#### Date Adjustment

The value of a date field can be adjusted by a certain time period, which is specified using the text:date-adjust attribute. If the time period is negative, it gets subtracted from the value of the date field, yielding a date before the current date.

The value of this attribute must conform to the time period format described in §3.2.6 of [xmlschema-2]. The value can be preceded by an optional minus sign to indicate a negative time duration.

<define name="text-date-attlist" combine="interleave">

<optional>

<attribute name="text:date-adjust">

<ref name="duration"/>

</attribute>

</optional>

</define>

### Time Fields

Time fields display the current time. They are very similar to the date fields described in section 6.2.1, supporting the same attributes except that for time fields, they are called text:time-value and text:time-adjust attributes.

This element contains the presentation of the time field value, depending on the data style specified. The default time is the current time. The value of this element can be preserved using the text:fixed attribute described in section 6.7.2.

<define name="paragraph-content" combine="choice">

<element name="text:time">

<ref name="text-time-attlist"/>

<text/>

</element>

</define>

The attributes that may be associated with the <text:time> element are:

* Time value
* Time adjustment
* Fixed (see section 6.7.2)
* Formatting style (see section 6.7.7). The formatting style must be a time data style, see section 14.7 for more information.

<define name="text-time-attlist" combine="interleave">

<interleave>

<ref name="common-field-fixed-attlist"/>

<ref name="common-field-data-style-name-attlist"/>

</interleave>

</define>

#### Time Value

The text:time-value attribute records the time at which the document was last edited.

Some applications support date and time in addition to date-only values.

The value of this attribute must conform with either the “dateTime” or “time” data types described in §3.2.7 and §3.2.8 of [xmlschema-2]. If no value is specified, the current time is assumed, even if the field is marked fixed.

<define name="text-time-attlist" combine="interleave">

<optional>

<attribute name="text:time-value">

<ref name="timeOrDateTime"/>

</attribute>

</optional>

</define>

#### Time Adjustment

The value of a time field can be adjusted by a certain time period, which is specified using the text:time-adjust attribute.

The value of this attribute must conform to the time period format described in §3.2.6 of [xmlschema-2]. The value can be preceded by an optional minus sign to indicate a negative time duration. Positive values adjust the time to a time in the future, while negative values adjust the time to a time in the past. The duration is truncated to full minutes.

<define name="text-time-attlist" combine="interleave">

<optional>

<attribute name="text:time-adjust">

<ref name="duration"/>

</attribute>

</optional>

</define>

**Example:** **Time adjust attributes and their effects**

If the attribute text:time-adjust="PTM15", the time field displays a time which is 15 minutes later than the actual time specified by the time field value.

If the attribute text:time-adjust="-PTH1", the time field displays a time which is one hour before the actual time specified by the time field value.

### Page Number Fields

Page number fields display the current page number. These fields are particularly useful in headers and footers. E.g., if a page number field is inserted into a footer, the current page number is displayed on every page on which the footer appears.

The attributes that may be associated with the <text:page-number> element are:

* Page adjustment
* Display previous or following page numbers
* Fixed (see section 6.7.2)
* Formatting style (see section 6.7.8)   
  Page numbers can be formatted according to the number format described in section 2.9. If a number style is not specified, the page numbers are formatted according to the number style defined in the current page style.

<define name="paragraph-content" combine="choice">

<element name="text:page-number">

<ref name="text-page-number-attlist"/>

<text/>

</element>

</define>

<define name="text-page-number-attlist" combine="interleave">

<interleave>

<ref name="common-field-num-format-attlist"/>

<ref name="common-field-fixed-attlist"/>

</interleave>

</define>

**Note:** To display the total number of pages in a document, use the <text:page-count/> field described in section 6.4.17.

#### Page Adjustment

The value of a page number field can be adjusted by a specified number, allowing the display of page numbers of following or preceding pages. The adjustment amount is specified using the text:page-adjust attribute. When this attribute is used, the application:

1. Adds the value of the attribute to the current page number.
2. Checks to see if the resulting page exists.
3. If the page exists, the number of that page is displayed.
4. If the page does not exist, the value of the page number field remains empty and no number is displayed.

<define name="text-page-number-attlist" combine="interleave">

<optional>

<attribute name="text:page-adjust">

<ref name="integer"/>

</attribute>

</optional>

</define>

#### Display Previous or Following Page Numbers

The text:select-page attribute is used to display the number of the previous or the following page rather than the number of the current page.

<define name="text-page-number-attlist" combine="interleave">

<optional>

<attribute name="text:select-page">

<choice>

<value>previous</value>

<value>current</value>

<value>next</value>

</choice>

</attribute>

</optional>

</define>

**Note:** To display the current page number on all pages except the first or last page, use a combination of the text:select page and text:page adjust attributes.

**Example: Displaying the current page number on all pages except the first page**

<text:page-number text:select-page="previous"  
 text:page-adjust="1"  
 style:num-format="1"/>

### Page Continuation Text

In some publications, a continuation reminder is printed at the bottom of the page in addition to the page number. To include a continuation reminder, use the <text:page-continuation> element.

<define name="paragraph-content" combine="choice">

<element name="text:page-continuation">

<ref name="text-page-continuation-attlist"/>

<text/>

</element>

</define>

The attributes associated with the <text:page-continuation> element are:

* Previous or following page
* String value

#### Previous or Following Page

This attribute specifies whether to check for a previous or next page and if the page exists, the continuation text is printed.

<define name="text-page-continuation-attlist" combine="interleave">

<attribute name="text:select-page">

<choice>

<value>previous</value>

<value>next</value>

</choice>

</attribute>

</define>

#### String Value

This attribute specifies the continuation text to display. If this attribute is omitted, the element content is used.

<define name="text-page-continuation-attlist" combine="interleave">

<optional>

<attribute name="text:string-value">

<ref name="string"/>

</attribute>

</optional>

</define>

### Sender Fields

There are several fields which contain information about the sender of the current document, for example, name and email address. The information about the sender is taken from the OpenDocument user information dialog. If a sender field is marked fixed using the text:fixed attribute, the original sender information in the sender fields is preserved. (cf. section 6.7.2) Otherwise, the information is updated each time the file is edited, causing the fields to change value when the document is edited by a different user.

#### First Name

This element represents the first name of the sender.

<define name="paragraph-content" combine="choice">

<element name="text:sender-firstname">

<ref name="common-field-fixed-attlist"/>

<text/>

</element>

</define>

#### Last Name

This element represents the last name of the sender.

<define name="paragraph-content" combine="choice">

<element name="text:sender-lastname">

<ref name="common-field-fixed-attlist"/>

<text/>

</element>

</define>

#### Initials

This element represents the initials of the sender.

<define name="paragraph-content" combine="choice">

<element name="text:sender-initials">

<ref name="common-field-fixed-attlist"/>

<text/>

</element>

</define>

#### Title

This element represents the title of the sender.

<define name="paragraph-content" combine="choice">

<element name="text:sender-title">

<ref name="common-field-fixed-attlist"/>

<text/>

</element>

</define>

#### Position

This element represents the position of the sender.

<define name="paragraph-content" combine="choice">

<element name="text:sender-position">

<ref name="common-field-fixed-attlist"/>

<text/>

</element>

</define>

#### Email Address

This element represents the email address of the sender.

<define name="paragraph-content" combine="choice">

<element name="text:sender-email">

<ref name="common-field-fixed-attlist"/>

<text/>

</element>

</define>

#### Private Telephone Number

This element represents the private telephone number of the sender.

<define name="paragraph-content" combine="choice">

<element name="text:sender-phone-private">

<ref name="common-field-fixed-attlist"/>

<text/>

</element>

</define>

#### Fax Number

This element represents the facsimile number of the sender.

<define name="paragraph-content" combine="choice">

<element name="text:sender-fax">

<ref name="common-field-fixed-attlist"/>

<text/>

</element>

</define>

#### Company Name

This element represents the name of the company that employs the sender.

<define name="paragraph-content" combine="choice">

<element name="text:sender-company">

<ref name="common-field-fixed-attlist"/>

<text/>

</element>

</define>

#### Office Telephone Number

This element represents the office telephone number of the sender.

<define name="paragraph-content" combine="choice">

<element name="text:sender-phone-work">

<ref name="common-field-fixed-attlist"/>

<text/>

</element>

</define>

#### Street

This element represents the street name of the address of the sender.

<define name="paragraph-content" combine="choice">

<element name="text:sender-street">

<ref name="common-field-fixed-attlist"/>

<text/>

</element>

</define>

#### City

This element represents the city name of the address of the sender.

<define name="paragraph-content" combine="choice">

<element name="text:sender-city">

<ref name="common-field-fixed-attlist"/>

<text/>

</element>

</define>

#### Postal Code

This element represents the postal code of the address of the sender.

<define name="paragraph-content" combine="choice">

<element name="text:sender-postal-code">

<ref name="common-field-fixed-attlist"/>

<text/>

</element>

</define>

#### Country

This element represents the country of the address of the sender.

<define name="paragraph-content" combine="choice">

<element name="text:sender-country">

<ref name="common-field-fixed-attlist"/>

<text/>

</element>

</define>

#### State or Province

This element represents the state or province of the address of the sender, if applicable.

<define name="paragraph-content" combine="choice">

<element name="text:sender-state-or-province">

<ref name="common-field-fixed-attlist"/>

<text/>

</element>

</define>

### Author Fields

There are two elements available to display the author of a document. One element displays the full name of the author and the other element displays the initials of the author.

The value of author fields can be fixed using the text:fixed attribute. Marking an author field as fixed preserves the original field content. Otherwise, the field content changes each time the document is updated, to reflect the last author of the document.

#### Name of the Author

This element represents the full name of the author.

<define name="paragraph-content" combine="choice">

<element name="text:author-name">

<ref name="common-field-fixed-attlist"/>

<text/>

</element>

</define>

#### Initials of the Author

This element represents the initials of the author.

<define name="paragraph-content" combine="choice">

<element name="text:author-initials">

<ref name="common-field-fixed-attlist"/>

<text/>

</element>

</define>

### Chapter Fields

Chapter fields display one of the following:

* The name of the current chapter
* The number of the current chapter
* Both the name and number of the current chapter

If the chapter field is placed inside a header or footer, it displays the current chapter name or number on every page.

<define name="paragraph-content" combine="choice">

<element name="text:chapter">

<ref name="text-chapter-attlist"/>

<text/>

</element>

</define>

The attributes that may be associated with the <text:chapter> element are:

* Display
* Outline level

#### Display

The text:display attribute specifies the information that the chapter field should display.

<define name="text-chapter-attlist" combine="interleave">

<attribute name="text:display">

<choice>

<value>name</value>

<value>number</value>

<value>number-and-name</value>

<value>plain-number-and-name</value>

<value>plain-number</value>

</choice>

</attribute>

</define>

**Example:** If the current chapter number is 2.4, the chapter title is Working with Tables, the prefix is [, and suffix is ], the possible display options and results are as follows:

| Value of text:display attribute | Field content displayed |
| --- | --- |
| number | [2.4] |
| name | Working with Tables |
| number-and-name | [2.4] Working with Tables |
| plain-number | 2.4 |
| plain-number-and-name | 2.4 Working with Tables |

#### Outline Level

This attribute is used to specify the outline level to use. The chapter field displays the chapter number or title up to the specified outline level.

<define name="text-chapter-attlist" combine="interleave">

<attribute name="text:outline-level">

<ref name="nonNegativeInteger"/>

</attribute>

</define>

### File Name Fields

File name fields display the name of the file that is currently being edited.

The attributes that may be associated with the <text:file-name> element are:

* Display
* Fixed

<define name="paragraph-content" combine="choice">

<element name="text:file-name">

<ref name="text-file-name-attlist"/>

<text/>

</element>

</define>

#### Display

The text:display attribute specifies how much of the file name to display. The following display options are allowed:

* The full file name including the path and the extension
* The file path only
* The file name only
* The file name and the extension

The filename might be an IRI, either because an IRI has been used to retrieve the file, or the application internally uses IRIs and therefore converts even system specific paths into an IRI. If this is the case, and if the path, the name or the extension cannot be evaluated from the IRI, then the IRI should be displayed unmodified.

<define name="text-file-name-attlist" combine="interleave">

<optional>

<attribute name="text:display">

<choice>

<value>full</value>

<value>path</value>

<value>name</value>

<value>name-and-extension</value>

</choice>

</attribute>

</optional>

</define>

#### Fixed File Name Fields

If a file name field is fixed, its value does not change when the file is edited.

<define name="text-file-name-attlist" combine="interleave">

<ref name="common-field-fixed-attlist"/>

</define>

### Document Template Name Fields

The document template name field displays information about the document template in use, such as the template title or the file name.

The only attribute that may be associated with the <text:template-name> element is:

* Display

<define name="paragraph-content" combine="choice">

<element name="text:template-name">

<ref name="text-template-name-attlist"/>

<text/>

</element>

</define>

#### Display

This attribute specifies which information about the document template to display. The following display options are allowed:

* The full file name including the path and the extension
* The file path only
* The file name only
* The file name and the extension
* The title
* The area of the document template

The latter two values can be used for template dialogs. The values are a superset of the display values available for the <text:file-name> element.

<define name="text-template-name-attlist">

<optional>

<attribute name="text:display">

<choice>

<value>full</value>

<value>path</value>

<value>name</value>

<value>name-and-extension</value>

<value>area</value>

<value>title</value>

</choice>

</attribute>

</optional>

</define>

### Sheet Name Fields

For Spreadsheet documents, sheet name fields display the name of the sheet that is currently being edited.

<define name="paragraph-content" combine="choice">

<element name="text:sheet-name">

<text/>

</element>

</define>

## Variable Fields

OpenDocument text documents can contain variables, which are processed or displayed using variable fields. A variable is a name/value pair. The variable name is used throughout the document to identify a particular variable, and therefore variable names cannot be reused for different types of variables. Most variable fields support different value types, such as numbers, dates, strings, and so on. In the OpenDocument file format, a variable must be declared at the beginning of a document.

There are three types of variables:

* **Simple variables**

Simple variables, usually called variables, can take different values at different positions throughout a document. Simple variables can be set using either setter or input fields. Setter fields contain an expression, which is used to compute the new value of the variable. Input fields prompt the user for the new value. Simple variables can be used to display different text in recurring elements, such as headers or footers.

* **User variables**

User variables have the same value throughout a document. If a user variable is set anywhere within the document, all fields in the document that display the user variable have the same value. In the office application user interface, a user variable can be set at any occurrence of a user field, or by using user variable input fields. In the OpenDocument file format, the value of the user variable can only be set after the variable is declared.

* **Sequence variables**

Sequence variables are used to number certain items in an OpenDocument text document, for example, images or tables.

Expression and text input fields are also variable fields, but they are not associated with any particular variables. Since their functionality is closely related to that of the variable fields, they are also described in this section of the manual.

Variables must be declared before they can be used. The variable declarations are collected in container elements for the particular variable type. The OpenDocument code for declaring variables is described in sections 6.3.1, 6.3.5 and 6.3.8.

### Declaring Simple Variables

Simple variables are declared using <text:variable-decl> elements. The declaration specifies the name and the value type of the variable.

To specify the name and value type of the simple variable, the following attributes are attached to the <text:variable-decl> element:

* text:name

The name of the variable must be unique. The name cannot already be used for any other type of variable. See section 6.7.3 for information on using this attribute.

* office:value-type

See section 6.7.1 for information on using this attribute.

<define name="text-variable-decl">

<element name="text:variable-decl">

<ref name="common-field-name-attlist"/>

<ref name="common-value-type-attlist"/>

</element>

</define>

### Setting Simple Variables

Simple variables can be set using variable setter elements. This element contains the presentation of the value of the variable, which can be empty if the text:display attribute is set to none.

The attributes that may be associated with the <text:variable-set> element are:

* text:name

This attribute specifies the name of the variable to set. It must match the name of a variable that has already been declared. See section 6.7.3 for information on using this attribute.

* text:formula

This attribute contains the formula to compute the value of the variable field. If the formula equals the content of the field element, this attribute can be omitted. See section 6.7.6 for information on using this attribute.

* office:value-type and the appropriate value attribute

See section 6.7.1 for information on using these attributes.

**Note:** A simple variable should not contain different value types at different places in a document. However, an implementation may allow the use of different value types for different instances of the same variable. In the case of the numeric value types float, percentage, and currency, the value is automatically converted to the different value type. For value types that are stored internally as numbers, such as date, time, and boolean types, the values are reinterpreted as numbers of the respective types. If a variable is used for both string and non-string types, the behavior is undefined, therefore this practice is not recommended.

* text:display

This attribute can be used to specify whether or not to display the value of the <text:variable-set> element. If the text:display attribute is set to value, the value of the variable is displayed. If the attribute is set to none, the value is not displayed. See section 6.7.5 for information on using this attribute.

* style:data-style-name

This attribute specifies the data style to use to format a numeric, Boolean, or date/time variable. If a data style is not specified, a standard data style is used. See section 6.7.7 for information on using this attribute.

<define name="paragraph-content" combine="choice">

<element name="text:variable-set">

<interleave>

<ref name="common-field-name-attlist"/>

<ref name="common-field-formula-attlist"/>

<ref name="common-value-and-type-attlist"/>

<ref name="common-field-display-value-none-attlist"/>

<ref name="common-field-data-style-name-attlist"/>

</interleave>

<text/>

</element>

</define>

### Displaying Simple Variables

The <text:variable-get> element reads and displays the value of a simple variable. The value of this element is the value of the last preceding <text:variable-set> element with an identical text:name attribute. The element determines how the value of the variable is presented, in accordance with the chosen formatting style.

The attributes that may be associated with the <text:variable-get> element are:

* text:name

This attribute specifies the name of the variable to display. The name must match the name of a preceding <text:variable-del> element. See section 6.7.3 for information on using this attribute.

* text:display

This attribute can be used to specify whether to display the formula for a simple variable or the computed value of the variable. See section 6.7.5 for information on using this attribute.

* style:data-style-name

This attribute specifies the data style to use to format a numeric, Boolean, or date/time variable. If a data style is not specified, a standard data style is used. See section 6.7.7 for information on using this attribute.

<define name="paragraph-content" combine="choice">

<element name="text:variable-get">

<interleave>

<ref name="common-field-name-attlist"/>

<ref name="common-field-display-value-formula-attlist"/>

<ref name="common-field-data-style-name-attlist"/>

</interleave>

<text/>

</element>

</define>

### Simple Variable Input Fields

As an alternative to setting simple variables using formulas in variable setter elements, the user can be prompted for variable values. To do this, use the <text:variable-input> element. This element contains the presentation of the variable's value according to the chosen formatting style. The presentation can be empty if the text:display attribute is set to none.

The attributes that may be associated with the <text:variable-input> element are:

* text:name

This attribute specifies the name of the variable to display. It must match the name of a variable that was already declared. See section 6.7.3 for information on using this attribute.

* text:description

This optional attribute contains a brief message that is presented to users when they are prompted for input. The message should give users enough information about the variable or the use of the value within the document to enable them to choose an appropriate value. See section 6.7.4 for information on using this attribute.

* office:value-type and the appropriate value attribute

See section 6.7.1 for information on using these attributes.

* text:display

This attribute can be used to specify whether to display or hide the value of the variable through the variable input field. See section 6.7.5 for information on using this attribute.

* style:data-style-name

This attribute specifies the data style to use to format a numeric, Boolean, or date/time variable. If a data style is not specified, a standard data style is used. See section 6.7.7 for information on using this attribute.

<define name="paragraph-content" combine="choice">

<element name="text:variable-input">

<interleave>

<ref name="common-field-name-attlist"/>

<ref name="common-field-description-attlist"/>

<ref name="common-value-type-attlist"/>

<ref name="common-field-display-value-none-attlist"/>

<ref name="common-field-data-style-name-attlist"/>

</interleave>

<text/>

</element>

</define>

### Declaring User Variables

User variables contain values that are displayed using appropriate fields. Unlike simple variables, user variables have the same value throughout a document. For this reason, the value of user variables is stored in the variable declaration itself.

The attributes that may be associated with the <text:user-field-del> element are:

* text:name

This attribute specifies the name of the variable to be declared. The name must be unique. It cannot already be used for any other type of variable including simple and sequence variables. See section 6.7.3 for information on using this attribute.

* text:formula

This attribute contains the formula to compute the value of the user variable field. If the formula is the same as the content of the field element, this attribute can be omitted. See section 6.7.6 for information on using this attribute.

* office:value-type and the appropriate value attribute

See section 6.7.1 for information on using these attributes.

<define name="text-user-field-decl">

<element name="text:user-field-decl">

<ref name="common-field-name-attlist"/>

<optional>

<ref name="common-field-formula-attlist"/>

</optional>

<ref name="common-value-and-type-attlist"/>

</element>

</define>

### Displaying User Variables

The content of user variables can be displayed using <text:user-field-get> elements.

The attributes that may be associated with the <text:user-field-get> element are:

* text:name

This attribute specifies the name of the variable to display. The name must match the name of a preceding <text:user-field-del> element. See section 6.7.3 for information on using this attribute.

* text:display

This attribute can be used to specify whether to:

* + Display the formula used to compute the value of the user variable.
  + Display the value of the user variable.
  + Hide the user variable fields.
    - See section 6.7.5 for information on using this attribute.

**Note:** Since the office application user interfaces usually allow users to edit a user field variable by clicking on any user field, a hidden <text:user-field-get> element can be used as an anchor to allow easy access to a particular user field variable.

* style:data-style-name

This attribute specifies the data style to use to format a numeric, Boolean, or date/time variable. If a data style is not specified, a standard data style is used. See section 6.7.7 for information on using this attribute.

<define name="paragraph-content" combine="choice">

<element name="text:user-field-get">

<interleave>

<ref name="common-field-name-attlist"/>

<ref name="common-field-display-value-formula-none-attlist"/>

<ref name="common-field-data-style-name-attlist"/>

</interleave>

<text/>

</element>

</define>

### User Variable Input Fields

An alternative method of setting user variables is to use input fields, similar to the input fields for simple variables. A user variable can be set in this way using the <text:user-field-input> element. Since the value of a user field variable is stored in the <text:user-field-del> element, the <text:user-field-input> element does not contain the value and value type attributes from the <text:variable-input> field.

The presentation can be empty if the text:display attribute is set to none.

The attributes that may be associated with the <text:user-field-input> element are:

* text:name

This attribute specifies the name of the variable to set. It must match the name of a variable that has already been declared. See section 6.7.3 for information on using this attribute.

* text:description

This optional attribute contains a brief message that is presented to users when they are prompted for input. The message should give users enough information about the variable or the use of the value within the document, to enable them to choose an appropriate value. See section 6.7.4 for information on using this attribute.

* style:data-style-name

This attribute specifies the data style to use to format a numeric, Boolean, or date/time variable. If a data style is not specified, a standard data style is used. See section 6.7.7 for information on using this attribute.

<define name="paragraph-content" combine="choice">

<element name="text:user-field-input">

<interleave>

<ref name="common-field-name-attlist"/>

<ref name="common-field-description-attlist"/>

<ref name="common-field-data-style-name-attlist"/>

</interleave>

<text/>

</element>

</define>

### Declaring Sequence Variables

Sequence variables are used to number items within an OpenDocument text document. Sequence variables are most commonly used for sequential numbering. However, expression formulas can be included in sequence fields to support more advanced sequences. See section 6.3.9 for more information on Using Sequence Fields and their uses.

Sequence variables are declared using the <text:sequence-del> element.

To facilitate chapter-specific numbering, attributes can be attached to a sequence variable to specify a chapter level and a separation character. The attributes that may be associated with the <text:sequence-del> element are:

* text:name

This attribute specifies the name of the variable to be declared. The name must be unique. It cannot already be used for any other type of variable including simple and user variables. See section 6.7.3 for information on using this attribute.

* text:display-outline-level

See section 6.3.8:Outline Level for information about this attribute.

* text:separation-character

See section 6.3.8:Separation Character for information about this attribute.

<define name="text-sequence-decl">

<element name="text:sequence-decl">

<ref name="text-sequence-decl-attlist"/>

</element>

</define>

<define name="text-sequence-decl-attlist" combine="interleave">

<ref name="common-field-name-attlist"/>

</define>

#### Outline Level

Sequences can be numbered by chapter. To use this feature, use the text:display-outline-level attribute to specify an outline level that determines which chapters to reference for the chapter-specific numbering. All chapters that are at or below the specified outline level reset the value of the sequence to zero, the default value. Also, the chapter number of the last chapter at or below the specified outline level is prefixed to the sequence number. Choosing an outline level of zero results in a straight sequence of all sequence elements for that sequence variable.

<define name="text-sequence-decl-attlist" combine="interleave">

<attribute name="text:display-outline-level">

<ref name="nonNegativeInteger"/>

</attribute>

</define>

#### Separation Character

If sequences are numbered by chapter, this attribute is used to choose a character to separate the chapter number from the sequence number.

If the value of the text:display-outline-level attribute is a non-zero value, a separation character may be specified. The default separation character is ".".Otherwise, if the value of text:display-outline-level is zero, this attribute must be omitted.

<define name="text-sequence-decl-attlist" combine="interleave">

<optional>

<attribute name="text:separation-character">

<ref name="character"/>

</attribute>

</optional>

</define>

**Example: Sequence variable**

The sequence variable 3.7.36#5 with a value of 5 is declared using:

| Attribute | Value |
| --- | --- |
| text:display-outline-level | 3 |
| text:separation-character | # |

### Using Sequence Fields

Once a sequence variable is declared, it can be used in sequence fields throughout the document. Most sequence fields simply increment and display the sequence variable. However, sequence fields can also assume a new start value at any given position in a document. This start value is computed using a formula which is contained in the sequence field. If a sequence field without a start value is added, the office application software automatically inserts an expression of the type variable+1.

Sequence fields are most commonly used for simple counting sequences. However, the ability to provide arbitrary expressions supports more complex sequences. To form a sequence of even numbers, all sequence elements for that particular variable need to contain a formula incrementing the value by two, for example, variable+2. A sequence with a starting value of 1 and all subsequent elements using the formula variable\*2 yields all powers of two. Since different sequence elements for the same sequence variable may contain different formulas, complex sequences may be constructed.

The attributes that may be associated with the <text:sequence> element are:

* text:name

This attribute specifies the name of the variable that the field is to display. It must match the name of a sequence variable that was already declared. See section 6.7.3 for information on using this attribute.

* text:formula

This optional attribute contains a formula to compute the value of the sequence field. If this attribute is omitted, an expression containing the content of the element is used. See section 6.7.6 for information on using this attribute.

* style:num-format and style:num-letter-sync

These attributes specify the numbering style to use. If a numbering style is not specified, the numbering style is inherited from the page style. See section 6.7.8 for information on these attributes.

* text:ref-name

See the section 6.3.9:Reference Name for more information about this attribute.

<define name="paragraph-content" combine="choice">

<element name="text:sequence">

<interleave>

<ref name="common-field-name-attlist"/>

<ref name="common-field-formula-attlist"/>

<ref name="common-field-num-format-attlist"/>

<ref name="text-sequence-ref-name"/>

</interleave>

<text/>

</element>

</define>

#### Reference Name

Sequence fields can be the target of references, as implemented using reference fields. See section 6.6.5 for more information about reference fields. To enable a reference field to identify a particular sequence field, the sequence field must contain an additional attribute containing a name. No two sequence fields can have the same reference name.

If the sequence field is not the target of a reference, this attribute can be omitted.

<define name="text-sequence-ref-name">

<optional>

<attribute name="text:ref-name">

<ref name="string"/>

</attribute>

</optional>

</define>

### Expression Fields

Expression fields contain expressions that are evaluated and the resulting value is displayed. The value of the expression is formatted according to the chosen formatting style.

The attributes that may be associated with the <text:expression> element are:

* text:formula

This attribute contains the actual expression used to compute the value of the expression field. See section 6.7.6 for information on using this attribute.

* office:value-type and the appropriate value attribute

See section 6.7.1 for information on using these attributes.

* text:display

Use this attribute to specify one of the following:

* + To display the value of the field.
  + To display the formula used to compute the value.

See section 6.7.5 for information on using this attribute.

* style:data-style-name

This attribute specifies the data style to use to format a numeric, Boolean, or date/time variable. If a data style is not specified, a standard data style is used. See section 6.7.7 for information on using this attribute.

<define name="paragraph-content" combine="choice">

<element name="text:expression">

<interleave>

<ref name="common-field-formula-attlist"/>

<optional>

<ref name="common-value-and-type-attlist"/>

</optional>

<ref name="common-field-display-value-formula-attlist"/>

<ref name="common-field-data-style-name-attlist"/>

</interleave>

<text/>

</element>

</define>

### Text Input Fields

A text input field is a variable field. From the point of view of the user interface, a text input field is similar to the <text:variable-input> and <text:user-field-input> fields. However, the text input field does not change the value of any variables.

The only attribute that may be associated with the <text:text-input> element is:

* text:description

This attribute contains a brief message that is presented to users when they are prompted for input. The message should give users enough information about the purpose of the field and how it is used within the document, to enable them to choose an appropriate value. See section 6.7.4 for information on using this attribute.

<define name="paragraph-content" combine="choice">

<element name="text:text-input">

<ref name="common-field-description-attlist"/>

<text/>

</element>

</define>

## Metadata Fields

Metadata fields display meta information about the document, such as, the document creation date or the time at which the document was last printed. The names of the metadata field elements correspond to the metadata elements described in Chapter 3.

All metadata field elements can be marked as fixed using the text:fixed attribute. (Cf. section 6.7.2)

Several metadata fields display a date or a time. The elements for these fields require an associated text:date-value or a text:time-value attribute, and optionally, they can also have a style:data-style-name attribute. See section 6.7.1 for more information on these attributes.

### Initial Creator

This element represents the name of the author who created the original document.

<define name="paragraph-content" combine="choice">

<element name="text:initial-creator">

<ref name="common-field-fixed-attlist"/>

<text/>

</element>

</define>

### Document Creation Date

This element represents the date on which the document was created.

<define name="paragraph-content" combine="choice">

<element name="text:creation-date">

<interleave>

<ref name="common-field-fixed-attlist"/>

<ref name="common-field-data-style-name-attlist"/>

<optional>

<attribute name="text:date-value">

<ref name="dateOrDateTime"/>

</attribute>

</optional>

</interleave>

<text/>

</element>

</define>

### Document Creation Time

This element represents the time at which the document was created.

<define name="paragraph-content" combine="choice">

<element name="text:creation-time">

<interleave>

<ref name="common-field-fixed-attlist"/>

<ref name="common-field-data-style-name-attlist"/>

<optional>

<attribute name="text:time-value">

<ref name="timeOrDateTime"/>

</attribute>

</optional>

</interleave>

<text/>

</element>

</define>

### Document Description

This element contains a brief description of the document.

<define name="paragraph-content" combine="choice">

<element name="text:description">

<ref name="common-field-fixed-attlist"/>

<text/>

</element>

</define>

### User-Defined Document Information

This element contains user-defined information about the document. It displays the information provided within a <meta:user-defined> element that has the same name.

<define name="paragraph-content" combine="choice">

<element name="text:user-defined">

<interleave>

<ref name="common-field-fixed-attlist"/>

<attribute name="text:name">

<ref name="string"/>

</attribute>

<ref name="common-field-data-style-name-attlist"/>

<optional>

<attribute name="office:value">

<ref name="double"/>

</attribute>

</optional>

<optional>

<attribute name="office:date-value">

<ref name="dateOrDateTime"/>

</attribute>

</optional>

<optional>

<attribute name="office:time-value">

<ref name="duration"/>

</attribute>

</optional>

<optional>

<attribute name="office:boolean-value">

<ref name="boolean"/>

</attribute>

</optional>

<optional>

<attribute name="office:string-value">

<ref name="string"/>

</attribute>

</optional>

</interleave>

<text/>

</element>

</define>

### Print Time

This element represents the time at which the document was last printed.

<define name="paragraph-content" combine="choice">

<element name="text:print-time">

<interleave>

<ref name="common-field-fixed-attlist"/>

<ref name="common-field-data-style-name-attlist"/>

<optional>

<attribute name="text:time-value">

<ref name="time"/>

</attribute>

</optional>

</interleave>

<text/>

</element>

</define>

### Print Date

This element represents the date on which the document was last printed.

<define name="paragraph-content" combine="choice">

<element name="text:print-date">

<interleave>

<ref name="common-field-fixed-attlist"/>

<ref name="common-field-data-style-name-attlist"/>

<optional>

<attribute name="text:date-value">

<ref name="date"/>

</attribute>

</optional>

</interleave>

<text/>

</element>

</define>

### Printed By

This element represents name of the last person who printed the document.

<define name="paragraph-content" combine="choice">

<element name="text:printed-by">

<ref name="common-field-fixed-attlist"/>

<text/>

</element>

</define>

### Document Title

This element represents the title of the document.

<define name="paragraph-content" combine="choice">

<element name="text:title">

<ref name="common-field-fixed-attlist"/>

<text/>

</element>

</define>

### Document Subject

This element represents the subject of the document.

<define name="paragraph-content" combine="choice">

<element name="text:subject">

<ref name="common-field-fixed-attlist"/>

<text/>

</element>

</define>

### Document Keywords

This element contains a list of keywords used to describe the document.

<define name="paragraph-content" combine="choice">

<element name="text:keywords">

<ref name="common-field-fixed-attlist"/>

<text/>

</element>

</define>

### Document Revision Number

This element contains the document revision number. When the document is created, the revision number is set to 1. Each time the document is saved, the document revision number is incremented.

<define name="paragraph-content" combine="choice">

<element name="text:editing-cycles">

<ref name="common-field-fixed-attlist"/>

<text/>

</element>

</define>

**Note:** Since the <text:editing-cycles> field can not be formatted, the revision number can be read from the element content. Therefore, no extra attribute is needed.

### Document Edit Duration

Every time a document is edited, the office application records the duration between the time the document is opened and the time the document is closed. It then adds the duration to an internal counter, thereby keeping track of the total time that has been spent editing the document.

<define name="paragraph-content" combine="choice">

<element name="text:editing-duration">

<interleave>

<ref name="common-field-fixed-attlist"/>

<ref name="common-field-data-style-name-attlist"/>

<optional>

<attribute name="text:duration">

<ref name="duration"/>

</attribute>

</optional>

</interleave>

<text/>

</element>

</define>

### Document Modification Time

This element represents the time at which the document was last modified.

This element displays the information from the <dc:date> element. The name was chosen to avoid confusion with <text:date> fields.

<define name="paragraph-content" combine="choice">

<element name="text:modification-time">

<interleave>

<ref name="common-field-fixed-attlist"/>

<ref name="common-field-data-style-name-attlist"/>

<optional>

<attribute name="text:time-value">

<ref name="time"/>

</attribute>

</optional>

</interleave>

<text/>

</element>

</define>

### Document Modification Date

This element represents the date on which the document was last modified.

This element displays the information from the <dc:date> element. The name was chosen to avoid confusion with <text:date> fields.

<define name="paragraph-content" combine="choice">

<element name="text:modification-date">

<interleave>

<ref name="common-field-fixed-attlist"/>

<ref name="common-field-data-style-name-attlist"/>

<optional>

<attribute name="text:date-value">

<ref name="date"/>

</attribute>

</optional>

</interleave>

<text/>

</element>

</define>

### Document Modified By

This element represents the name of the person who last modified the document.

<define name="paragraph-content" combine="choice">

<element name="text:creator">

<ref name="common-field-fixed-attlist"/>

<text/>

</element>

</define>

### Document Statistics Fields

These fields display how many objects of a certain type a document contains. They can be used to display the number of

* pages,
* paragraphs,
* words,
* characters,
* tables,
* images, or
* embedded objects.

<define name="paragraph-content" combine="choice">

<element>

<choice>

<name>text:page-count</name>

<name>text:paragraph-count</name>

<name>text:word-count</name>

<name>text:character-count</name>

<name>text:table-count</name>

<name>text:image-count</name>

<name>text:object-count</name>

</choice>

<ref name="common-field-num-format-attlist"/>

<text/>

</element>

</define>

## Database Fields

Documents can reference databases and display database information as text content. To display database information, the OpenDocument schema uses a group of text fields, collectively called database fields. Office applications may use database tables from SQL servers, therefore database fields can be used to access any SQL database, provided that the appropriate drivers are available.

A database may contain the following components:

* Tables, which store the actual data.
* Queries, which extract a subset of data from one or more tables.
* Forms, which present the data.
* Reports, which summarize the database content.

Database forms and reports are not relevant to text content, therefore they are not discussed in this chapter. From the point of view of embedding database information in OpenDocument text documents, queries and tables are considered the same. Therefore for the remainder of this section, the phrase *database table* refers to both database tables and database queries.

Database fields alone do not retrieve information from a database. In addition to the database fields, a set of database rows is also added to the document. When new data is added to the document, all database fields belonging to the added database table are updated. Using the office application user interface, database rows can be added in one of the following ways:

* Manually, using a data source browser and the data to fields function.
* Using the Form Letter menu item on the File menu. This menu item adds each row in the chosen data set into a newly created copy of the form letter.

To display data from a database table use the <text:database-display> element. The <text:database-select> and <text:database-next> elements can be used to determine which row within the current selection should be displayed. The current row number for a particular table can be displayed using the <text:database-row-number> element. Finally, the <text:database-name> field displays the name of the most recently used database, which is the address book file database by default.

### Database Field Data Source

A database field's source can either be the name of a database, or an IRI containing database connection resource data. If the source is a database name, then this name is used by all of the office application components to identify a database. All database fields contain a database name or connection resource, and most database fields also contain the name of a database table, which must be stored in the database. An additional attribute determines whether the database table refers to an SQL table, an OpenDocument query, or the result of a SQL command.

<define name="common-field-database-table">

<ref name="common-field-database-table-attlist"/>

<ref name="common-field-database-name"/>

</define>

#### Database Name

The text:database-name attribute specifies the source database by its name.

<define name="common-field-database-name" combine="choice">

<optional>

<attribute name="text:database-name">

<ref name="string"/>

</attribute>

</optional>

</define>

#### Connection Resource

The <form:connection-resource> element specifies the source database by an [XLink]. Its xlink:href attribute either references a file containing a database, or it contains information on how to make a connection to a database, for instance a [JDBC] URL. See also section 11.1.20.

<define name="common-field-database-name" combine="choice">

<ref name="form-connection-resource"/>

</define>

#### Database Table Name

The text:table-name attribute specifies a table within the source database.

<define name="common-field-database-table-attlist" combine="interleave">

<attribute name="text:table-name">

<ref name="string"/>

</attribute>

</define>

#### Database Type

The text:table-type attribute determines whether the database table refers to an SQL table, an OpenDocument query, or the result of a SQL command.

<define name="common-field-database-table-attlist" combine="interleave">

<optional>

<attribute name="text:table-type">

<choice>

<value>table</value>

<value>query</value>

<value>command</value>

</choice>

</attribute>

</optional>

</define>

### Displaying Database Content

The <text:database-display> element displays data from a database. When a new data set is added to a document, all fields that display data from that database table update their content.

The attributes that may be associated with the <text:database-display> element are:

* text:database-name, text:table-name and text:table-type

These attributes specify the database and database table that this field uses.

* text:database-column-name

See section 6.5.2:Column Name for information about this attribute.

* style:data-style-name

If the column specifies a numeric, Boolean, date, or time value, the data is formatted according to the appropriate data style. If no data style is specified, the data style assigned to this column in is used. See section 6.7.7 for more information about using this attribute.

<define name="paragraph-content" combine="choice">

<element name="text:database-display">

<ref name="text-database-display-attlist"/>

<text/>

</element>

</define>

<define name="text-database-display-attlist" combine="interleave">

<ref name="common-field-database-table"/>

</define>

<define name="text-database-display-attlist" combine="interleave">

<ref name="common-field-data-style-name-attlist"/>

</define>

#### Column Name

The text:column-name attribute specifies the column from which to display the data. The value of this attribute must be a column contained in the specified database.

<define name="text-database-display-attlist" combine="interleave">

<attribute name="text:column-name">

<ref name="string"/>

</attribute>

</define>

### Selecting the Next Database Row

The <text:database-next> element changes the row in the current selection which is used for display in all following <text:database-display> fields. The next row from the current selection is chosen if it satisfies a given condition. If the next row is wanted regardless of any condition, the condition may be omitted or set to true.

The attributes that may be associated with the <text:database-next> are:

* text:database-name, text:table-name and text:table-type

These attributes specify the database and the database table that this field uses.

* text:condition

See section 6.5.3:Condition for information about this attribute.

<define name="paragraph-content" combine="choice">

<element name="text:database-next">

<ref name="text-database-next-attlist"/>

</element>

</define>

<define name="text-database-next-attlist" combine="interleave">

<ref name="common-field-database-table"/>

</define>

#### Condition

The text:condition attribute specifies the condition expression. The expression is evaluated and if the result interpreted as a Boolean value is true, the next row is used as the new current row. Database field values can be used in the expression by enclosing in square brackets the database name, the table name, and the column name, separated by dots.

If the text:condition attribute is not present, it is assumes that the formula true, meaning that the next row is selected unconditionally.

<define name="text-database-next-attlist" combine="interleave">

<optional>

<attribute name="text:condition">

<ref name="formula"/>

</attribute>

</optional>

</define>

**Example:**

text:formula='ooo-w:[address book file.address.FIRSTNAME] == "Julie"'

This example specifies a condition that is true if the current row from an address book database table is the address for a person named Julie. If the condition shown in this example is used in a <text:database-next> element, the following happens:

* The <text:database-display> elements display the data from the first row of the current selection.
* If the FIRSTNAME column of the current row reads Julie, the current row is changed. Otherwise, nothing happens.
* If the first row is Julie, the following <text:database-display> elements display data from the second row. Otherwise, they display data from the first row.

See section 6.7.6 for more information on the formula syntax of a text:condition attribute, which is the same as that of the text:formula attribute.

### Selecting a Row Number

The <text:database-row-select> element selects a specific row from the current selection. As with the <text:database-row-next> element, a condition can be specified so that the given row is only selected if the condition is true.

The attributes that may be associated with the <text:database-row-select> are:

* text:database-name, text:table-name and text:table-type

These attributes determine the database and the database table that this field uses.

* text:condition

This attribute specifies the condition expression. See section 6.5.3 for a full explanation of how to use this attribute.

* text:row-number

See the following section 6.5.4:Selecting the Row Number about this attribute.

<define name="paragraph-content" combine="choice">

<element name="text:database-row-select">

<ref name="text-database-row-select-attlist"/>

</element>

</define>

<define name="text-database-row-select-attlist" combine="interleave">

<ref name="common-field-database-table"/>

</define>

<define name="text-database-row-select-attlist" combine="interleave">

<optional>

<attribute name="text:condition">

<ref name="formula"/>

</attribute>

</optional>

</define>

#### Selecting the Row Number

This attribute specifies the row number to select when a condition is true.

<define name="text-database-row-select-attlist" combine="interleave">

<optional>

<attribute name="text:row-number">

<ref name="nonNegativeInteger"/>

</attribute>

</optional>

</define>

### Displaying the Row Number

The <text:database-row-number> element displays the current row number for a given table. Note that the element displays the actual row number from the database and not the row number of the current selection that is used as an attribute value in the <text:database-row-select> element.

The attributes that may be associated with the <text:database-row-number> are:

* text:database-name, text:table-name and text:table-type

These attributes determine the database and the database table that this field uses.

* style:num-format and style:num-letter-sync

These attributes determine how the number should be formatted. See section 6.7.8 for more information on how to use these attributes.

* text:value

This attribute specifies the current row number. The number changes when new data is added to the current document.

<define name="paragraph-content" combine="choice">

<element name="text:database-row-number">

<interleave>

<ref name="common-field-database-table"/>

<ref name="common-field-num-format-attlist"/>

<optional>

<attribute name="text:value">

<ref name="nonNegativeInteger"/>

</attribute>

</optional>

</interleave>

<text/>

</element>

</define>

### Display Current Database and Table

Office applications may keeps track of the last database and table that was used in the document. In other words, the table that is used by the last field that was inserted into the document. The <text:database-name> element displays the database and table name of the most recently used table.

The attributes that may be associated with the <text:database-name> element are:

* text:database-name, text:table-name and text:table-type

These attributes determine the database and the database table that this field uses.

<define name="paragraph-content" combine="choice">

<element name="text:database-name">

<ref name="common-field-database-table"/>

<text/>

</element>

</define>

## More Fields

### Page Variable Fields

Page variables allow an alternative page numbering scheme to be defined. There is only one page variable, and it is set by any set page variable field in the document. The value of the page variable is increased on each page, in the same way as regular page numbers.

#### Setting Page Variable Fields

To set a page variable field, use the <text:variable-page-set> element.

<define name="paragraph-content" combine="choice">

<element name="text:page-variable-set">

<ref name="text-set-page-variable-attlist"/>

<text/>

</element>

</define>

#### Turning Page Variables On or Off

At the beginning of a document, the page variable is inactive. The text:active attribute can be used to disable a page variable after it was used in the document.

<define name="text-set-page-variable-attlist" combine="interleave">

<optional>

<attribute name="text:active">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Page Variable Adjustment

The text:page-adjust attribute determines the page adjustment. The value of the active page variable is the current page number plus the closest page adjustment value that was previously set.

<define name="text-set-page-variable-attlist" combine="interleave">

<optional>

<attribute name="text:page-adjust">

<ref name="integer"/>

</attribute>

</optional>

</define>

#### Displaying Page Variable Fields

The <text:variable-page-get> element displays the value of the page variable. The field can be formatted in the same way as regular page number fields.

<define name="paragraph-content" combine="choice">

<element name="text:page-variable-get">

<ref name="text-get-page-variable-attlist"/>

<text/>

</element>

</define>

The attributes that may be associated with the <text:get-page-variable> element are:

* style:num-format and style:num-letter-sync

These attributes determine how the number should be formatted. See section 6.7.8 for more information on how to use these attributes.

<define name="text-get-page-variable-attlist" combine="interleave">

<ref name="common-field-num-format-attlist"/>

</define>

### Placeholders

The OpenDocument format uses placeholder fields to indicate locations in a document where the user must fill in some information. For example in a letter template, a section of the document can be reserved for the address of the recipient. A placeholder field displays text informing the user about the purpose of the placeholder and sometimes includes a description. Placeholder fields can represent different text elements, such as text or tables.

This element contains some brief text which is displayed with the placeholder.

<define name="paragraph-content" combine="choice">

<element name="text:placeholder">

<ref name="text-placeholder-attlist"/>

<text/>

</element>

</define>

The attributes that may be associated with the <text:placeholder> element are:

* Placeholder type
* Placeholder description

#### Placeholder Type

There are five different types of placeholder, representing the five possible types of content: text, tables, text boxes, images, or objects. The text:placeholder-type attribute represents the content type. This attribute is mandatory and it indicates which type of text content the placeholder represents. The value of the attribute can be text, text-box, image, table, or object.

<define name="text-placeholder-attlist" combine="interleave">

<attribute name="text:placeholder-type">

<choice>

<value>text</value>

<value>table</value>

<value>text-box</value>

<value>image</value>

<value>object</value>

</choice>

</attribute>

</define>

#### Placeholder Description

In addition to the brief text stored in the element content, may be associated a text:description attribute with the placeholder element. This attribute is optional. The purpose of the attribute is to contain a more elaborate description of the purpose of the placeholder than the description stored in the element content. See section 6.7.4 for information on using the text:description attribute.

<define name="text-placeholder-attlist" combine="interleave">

<ref name="common-field-description-attlist"/>

</define>

### Conditional Text Fields

Text fields can be used to display one text or another, depending on a condition. Conditional text fields are given a condition and two text strings. If the condition is true, one of the text strings is displayed. If the condition is false, the other text string is displayed.

<define name="paragraph-content" combine="choice">

<element name="text:conditional-text">

<ref name="text-conditional-text-attlist"/>

<text/>

</element>

</define>

The attributes that may be associated with the <text:conditional-text> element are:

* Condition
* Text to display if the condition is true
* Text to display if the condition is false
* Current condition

The text:condition attribute contains a Boolean expression. Depending on the result, the value of the text:display-if-true or text:display-if-false attribute is displayed.

<define name="text-conditional-text-attlist" combine="interleave">

<attribute name="text:condition">

<ref name="formula"/>

</attribute>

</define>

#### Text to Display if the Condition is True

The text:string-value-if-true attribute contains the text string to display if the condition is true.

<define name="text-conditional-text-attlist" combine="interleave">

<attribute name="text:string-value-if-true">

<ref name="string"/>

</attribute>

</define>

#### Text to Display if the Condition is False

The text:string-value-if-false attribute contains the text string to display if the condition is false.

<define name="text-conditional-text-attlist" combine="interleave">

<attribute name="text:string-value-if-false">

<ref name="string"/>

</attribute>

</define>

#### Current Value and Condition

The text:current-value attribute contains the evaluation result of the condition given by the expression in the text:condition attribute. Explicitly giving the result allows applications to delay evaluating the result until necessary. This attribute is valuable for the following reasons:

* If the expression is costly to evaluate, for example, the expression contains references to several databases.
* To allow transformations to correctly display the state of the document without having to parse and evaluate the condition.

<define name="text-conditional-text-attlist" combine="interleave">

<optional>

<attribute name="text:current-value">

<ref name="boolean"/>

</attribute>

</optional>

</define>

**Note**: The value of this attribute is overwritten with a new value as soon as the application evaluates the expression. This attribute has no function other than to ease transformation or initially display the document.

### Hidden Text Field

The hidden text field is closely related to the conditional text field. It displays fixed text, except when the condition is true when it does not display anything.

<define name="paragraph-content" combine="choice">

<element name="text:hidden-text">

<ref name="text-hidden-text-attlist"/>

<text/>

</element>

</define>

The attributes that may be associated with the <text:hidden-text> element are:

* Condition
* Text
* Is hidden

#### Condition

The text:condition attribute contains a Boolean expression. If the expression evaluates to true, the text is hidden.

<define name="text-hidden-text-attlist" combine="interleave">

<attribute name="text:condition">

<ref name="formula"/>

</attribute>

</define>

#### Text

The text:string-value attribute specifies the text to display if the condition is false.

<define name="text-hidden-text-attlist" combine="interleave">

<attribute name="text:string-value">

<ref name="string"/>

</attribute>

</define>

#### Is Hidden

The text:is-hidden attribute specifies whether or not the field is currently visible. The purpose of this attribute is similar to that of the text:current-value attribute in the text:condition field. Recording the result allows transformations to correctly represent the document without having to parse the condition expression or evaluate the condition when loading the document.

<define name="text-hidden-text-attlist" combine="interleave">

<optional>

<attribute name="text:is-hidden">

<ref name="boolean"/>

</attribute>

</optional>

</define>

**Note**: The value of this attribute is overwritten with a new value as soon as the application evaluates the expression. This attribute has no function other than to ease transformation or initially display the document.

### Reference Fields

The OpenDocument format uses four types of reference field and each type is represented by its own element. The reference field types are based on the type of element they refer to; notes, bookmarks, references, and sequences. Every reference contains a reference format which determines what information about the referenced target is displayed. For example, references can display:

* The page number of the referenced target
* The chapter number of the referenced target
* Wording indicating whether the referenced target is above or below the reference field

In addition, each reference field must identify its target which is usually done using a name attribute. Bookmarks and references are identified by the name of the respective bookmark or reference. Footnotes, endnotes, and sequences are are assigned names by the application used to create the OpenDocument file format automatically.

<define name="paragraph-content" combine="choice">

<element>

<choice>

<name>text:reference-ref</name>

<name>text:bookmark-ref</name>

</choice>

<interleave>

<ref name="text-common-ref-content"/>

<ref name="text-ref-content"/>

</interleave>

</element>

</define>

<define name="paragraph-content" combine="choice">

<element name="text:note-ref">

<interleave>

<ref name="text-common-ref-content"/>

<ref name="text-note-ref-content"/>

<ref name="text-ref-content"/>

</interleave>

</element>

</define>

<define name="paragraph-content" combine="choice">

<element name="text:sequence-ref">

<interleave>

<ref name="text-common-ref-content"/>

<ref name="text-sequence-ref-content"/>

</interleave>

</element>

</define>

<define name="text-common-ref-content" combine="interleave">

<text/>

</define>

The attributes that may be associated with the reference field elements are:

* Reference name
* Reference format

#### Reference Name

The text:ref-name attribute identifies the referenced element. Since bookmarks and references have a name, this name is used by the respective reference fields. Footnotes, endnotes, and sequences are are identified by a name that is usually generated automatically.

<define name="text-common-ref-content" combine="interleave">

<optional>

<attribute name="text:ref-name">

<ref name="string"/>

</attribute>

</optional>

</define>

#### Note Class

For <text:note-ref> elements, the text:note-class attribute determines whether the field references a foot- or an endnote.

<define name="text-note-ref-content" combine="interleave">

<ref name="text-note-class"/>

</define>

#### Reference Format

The text:reference-format attribute determines what information about the reference is displayed. If the reference format is not specified, the page format is used as the default.

All types of reference fields support the following values for this attribute formats:

* page, which displays the number of the page on which the referenced item appears.
* chapter, which displays the number of the chapter in which the referenced item appears.
* direction, which displays whether the referenced item is above or below the reference field.
* text, which displays the text of the referenced item.

References to sequence fields support the following three additional values:

* category-and-value, which displays the name and value of the sequence.
* caption, which displays the caption in which the sequence is used.
* value, which displays the value of the sequence.

<define name="text-ref-content" combine="interleave">

<optional>

<attribute name="text:reference-format">

<choice>

<value>page</value>

<value>chapter</value>

<value>direction</value>

<value>text</value>

</choice>

</attribute>

</optional>

</define>

<define name="text-sequence-ref-content" combine="interleave">

<optional>

<attribute name="text:reference-format">

<choice>

<value>page</value>

<value>chapter</value>

<value>direction</value>

<value>text</value>

<value>category-and-value</value>

<value>caption</value>

<value>value</value>

</choice>

</attribute>

</optional>

</define>

**Example: Different reference formats and displays**

The following table shows all possible reference formats and the resulting reference display that can be used to refer to the table itself. The left column lists the value of the text:reference-format attribute and the right column

| Reference format | Reference display |
| --- | --- |
| page | 196 |
| chapter | 3.7.27 |
| text | Table 2: Examples of reference formats |
| direction | above |
| category-and-value | Table 1 |
| caption | Examples of reference formats |
| value | 1 |

### Script Fields

A script field stores scripts or sections of scripts. The field can be used to store and edit scripts that are attached to the document. The primary purpose of this field is to provide an equivalent to the <script> element in [HTML4], so that the content of a <script> element in HTML can be imported, edited, and exported using an office application software.

The source code for the script can be stored in one of the following ways:

* The <text:script> element contains the source code.
* The source code is stored in an external file. Use the xlink:href attribute to specify the location of the source file.

The element should have either a xlink:href attribute or content, but not both.

<define name="paragraph-content" combine="choice">

<element name="text:script">

<interleave>

<choice>

<group>

<attribute name="xlink:href">

<ref name="anyURI"/>

</attribute>

<optional>

<attribute name="xlink:type" a:defaultValue="simple">

<value>simple</value>

</attribute>

</optional>

</group>

<text/>

</choice>

<optional>

<attribute name="script:language">

<ref name="string"/>

</attribute>

</optional>

</interleave>

</element>

</define>

#### Script URL

The xlink:href attribute specifies the location of the file that contains the script source code. The script field should have either an URL attribute or content, but not both.

#### Script Language

The script:language attribute specifies the language in which the script source code is written, for example, JavaScript.

### Macro Fields

The macro field contains the name of a macro that is executed when the field is activated. The field also contains a description that is displayed as the field content.

The only attribute that may be associated with the <text:execute-macro> element is:

* Macro name

<define name="paragraph-content" combine="choice">

<element name="text:execute-macro">

<optional>

<attribute name="text:name">

<ref name="string"/>

</attribute>

</optional>

<optional>

<ref name="office-event-listeners"/>

</optional>

<text/>

</element>

</define>

#### Macro Name

The text:name attribute specifies the macro to invoke when the field is activated.

### Hidden Paragraph Fields

The hidden paragraph field has a similar function to the hidden text field. However, the hidden paragraph field does not have any content. It hides the paragraph in which it is contained. This allows a paragraph of formatted text to be hidden or displayed depending on whether a condition is true or false.

Hidden paragraph fields are often used together with form letters. For example, if a condition depends on a database field, a hidden paragraph field can be used to selectively include paragraphs in the form letter depending on the database content. Multiple paragraph fields can be contained one paragraph. The paragraph is displayed if the condition associated with at least one hidden paragraph field is false. Alternatively, the conditions associated with several hidden paragraph fields can be combined into a single condition for a single field using logical operations on the conditions.

**Note**: Unlike most fields, this field does not display text, but it affects the entire paragraph in which it is contained.

The attributes that may be associated with the <text:hidden-paragraph> element are:

* Condition
* Is hidden

<define name="paragraph-content" combine="choice">

<element name="text:hidden-paragraph">

<ref name="text-hidden-paragraph-attlist"/>

<text/>

</element>

</define>

#### Condition

The text:condition attribute contains a Boolean expression. If the condition is true, the paragraph is hidden. If the condition is false, the paragraph is displayed.

<define name="text-hidden-paragraph-attlist" combine="interleave">

<attribute name="text:condition">

<ref name="formula"/>

</attribute>

</define>

#### Is Hidden

The text:is-hidden attribute records whether the paragraph is currently visible or not. It has the same purpose as the corresponding attribute of the hidden text field, namely to allow correct display of the paragraph without having to evaluate the condition first. The value of this attribute is overwritten with a new value as soon as the application evaluates the expression.

**Note**: This attribute has no function other than to ease transformation or initially display the document.

<define name="text-hidden-paragraph-attlist" combine="interleave">

<optional>

<attribute name="text:is-hidden">

<ref name="boolean"/>

</attribute>

</optional>

</define>

### DDE Connection Fields

A DDE field allows information from a DDE connection to be displayed. The only parameter required for the DDE field is the name of the DDE connection that supplies the data to this field. This DDE connection element specifies the actual DDE field that appears in the text body.

The field element contains the content of the most recent data that was received from the DDE connection. This may be used to render the document if the DDE connection cannot be accessed.

See section 12.6 for the use of DDE connections.

<define name="paragraph-content" combine="choice">

<element name="text:dde-connection">

<attribute name="text:connection-name">

<ref name="string"/>

</attribute>

<text/>

</element>

</define>

The only attribute that may be associated with the <text:dde-connection> element is:

* DDE connection name

#### DDE Connection Name

The text:name attribute specifies the name of the DDE connection to which the field refers.

### Measure Fields

Within the text contained in measure drawing objects (see section 9.2.11), a <text:measure> field displays the current measure. The draw:kind attribute specifies which part of the measure is displayed. It my have one of the following values:

* value: The measure's value is displayed, for instance “12”
* unit: The measure's unit is displayed, for instance “inch”
* gap: A gap or blank is displayed if and only if the measure text's writing direction is perpendicular to the measure line. The purpose of this value is add some space between the measure line and the text if the text is displayed perpendicular to the measure line.

<define name="paragraph-content" combine="choice">

<element name="text:measure">

<attribute name="text:kind">

<choice>

<value>value</value>

<value>unit</value>

<value>gap</value>

</choice>

</attribute>

<text/>

</element>

</define>

### Table Formula Field

The table formula field is a legacy from previous versions of current office applications. It should not be used in new documents. It stores a formula to be used in tables, a function that is better performed by the table:formula attribute of the table cell.

**Note**: This element should not be used in new documents.

The table formula field can take the following attributes:

* text:formula

This attribute contains the actual expression used to compute the value of the table formula field. See section 6.7.6 for information on using this attribute.

* text:display

Use this attribute to specify one of the following:

* + To display the value of the field.
  + To display the formula used to compute the value.

See section 6.7.5 for information on using this attribute.

* style:data-style-name

This attribute specifies the data style to use to format a numeric, Boolean, or date/time variable. If a data style is not specified, a standard data style is used. See section 6.7.7 for information on using this attribute.

<define name="paragraph-content" combine="choice">

<element name="text:table-formula">

<interleave>

<ref name="common-field-formula-attlist"/>

<ref name="common-field-display-value-formula-attlist"/>

<ref name="common-field-data-style-name-attlist"/>

</interleave>

<text/>

</element>

</define>

## Common Field Attributes

The attributes described in this section can be used with several field elements.

### Variable Value Types and Values

Variables and most variable fields have a current value. Every variable has a value type that must be specified when the field supports multiple value types. The value type is specified using the office:value-type attribute.

<define name="common-value-type-attlist">

<attribute name="office:value-type">

<ref name="valueType"/>

</attribute>

</define>

Depending on the value type, the value itself is written to different value attributes. The supported value types, their respective value attributes, and how the values are encoded are described in the following table:

| Value Type | Value Attribute(s) | Encoded as... | Example |
| --- | --- | --- | --- |
| float | office:value | Numeric value | "12.345" |
| percentage | office:value | Numeric value | "0.50" |
| currency | office:value and  office:currency | Numeric value and  currency symbol | "100" "USD" |
| date | office:date-value | Date value as specified in §3.2.9 of [xmlschema-2], or date and time value as specified in §3.2.7 of [xmlschema-2] | "2003-04-17" |
| time | office:time-value | Duration, as specified in §3.2.6 of [xmlschema-2] | "PT03H30M00S" |
| boolean | office:boolean-value | true or false | "true" |
| string | office:string-value | Strings | "abc def" |

The OpenDocument concept of field values and value types and their encoding in XML is modeled on the corresponding XML for table cell attributes. See section 8.1.3 for information on table cells and their attributes.

The definition of the entity %value-attlist; is as follows:

<define name="common-value-and-type-attlist">

<choice>

<group>

<attribute name="office:value-type">

<value>float</value>

</attribute>

<attribute name="office:value">

<ref name="double"/>

</attribute>

</group>

<group>

<attribute name="office:value-type">

<value>percentage</value>

</attribute>

<attribute name="office:value">

<ref name="double"/>

</attribute>

</group>

<group>

<attribute name="office:value-type">

<value>currency</value>

</attribute>

<attribute name="office:value">

<ref name="double"/>

</attribute>

<optional>

<attribute name="office:currency">

<ref name="string"/>

</attribute>

</optional>

</group>

<group>

<attribute name="office:value-type">

<value>date</value>

</attribute>

<attribute name="office:date-value">

<ref name="dateOrDateTime"/>

</attribute>

</group>

<group>

<attribute name="office:value-type">

<value>time</value>

</attribute>

<attribute name="office:time-value">

<ref name="duration"/>

</attribute>

</group>

<group>

<attribute name="office:value-type">

<value>boolean</value>

</attribute>

<attribute name="office:boolean-value">

<ref name="boolean"/>

</attribute>

</group>

<group>

<attribute name="office:value-type">

<value>string</value>

</attribute>

<optional>

<attribute name="office:string-value">

<ref name="string"/>

</attribute>

</optional>

</group>

</choice>

</define>

### Fixed

The text:fixed attribute specifies whether or not the value of a field element is fixed. If the value of a field is fixed, the value of the field element to which this attribute is attached is preserved in all future edits of the document. If the value of the field is not fixed, the value of the field may be replaced by a new value when the document is edited.

This attribute can be used with:

* Date fields
* Time fields
* Page number fields
* All sender fields
* All author fields

<define name="common-field-fixed-attlist">

<optional>

<attribute name="text:fixed">

<ref name="boolean"/>

</attribute>

</optional>

</define>

### Variable Name

Use the text:name attribute to specify the name of a variable when it is being declared, set, or displayed a variable. This attribute can be used with any of the following elements:

* <text:variable-del>
* <text:variable-set>
* <text:variable-get>
* <text:variable-input>
* <text:user-field-del>
* <text:user-field-get>
* <text:user-field-input>
* <text:sequence-del>
* <text:sequence>

When this attribute is being used to specify the name of a variable to display, a variable of the appropriate type with the same name must already have been declared.

<define name="common-field-name-attlist">

<attribute name="text:name">

<ref name="variableName"/>

</attribute>

</define>

### Description

The text:description attribute contains a brief message that is displayed when users are prompted for input. This attribute can be used with any of the following elements:

* <text:placeholder>
* <text:variable-input>
* <text:user-field-input>
* <text:text-input>

<define name="common-field-description-attlist">

<optional>

<attribute name="text:description">

<text/>

</attribute>

</optional>

</define>

### Display

The text:display attribute supports up to three values as follows:

* value  
  This value displays the value of the field. Some fields do not support this value. In these cases, the text:display attribute only takes the values value or none, and value or formula, respectively.
* formula  
  This value allows the display of the formula rather than the value of the field. Some fields do not support this value. In these cases, the text:display attribute only takes the values value or none, and value or formula, respectively.
* none  
  Several variable fields support this value, which hides the field content. This allows variables to be set in one part of the document and displayed in another part of the document.

This attribute can be used with any of the following elements:

* <text:variable-set>
* <text:variable-get>
* <text:variable-input>
* <text:user-field-get>
* <text:expression>

<define name="common-field-display-value-none-attlist">

<optional>

<attribute name="text:display">

<choice>

<value>value</value>

<value>none</value>

</choice>

</attribute>

</optional>

</define>

<define name="common-field-display-value-formula-none-attlist">

<optional>

<attribute name="text:display">

<choice>

<value>value</value>

<value>formula</value>

<value>none</value>

</choice>

</attribute>

</optional>

</define>

<define name="common-field-display-value-formula-attlist">

<optional>

<attribute name="text:display">

<choice>

<value>value</value>

<value>formula</value>

</choice>

</attribute>

</optional>

</define>

### Formula

The text:formula attribute contains the formula or expression used to compute the value of the field. This attribute can be used with any of the following elements:

* <text:variable-set>
* <text:user-field-del>
* <text:sequence>
* <text:expression>

The formula should start with a namespace prefix that indicates the syntax and semantic used within the formula.

<define name="common-field-formula-attlist">

<optional>

<attribute name="text:formula">

<ref name="formula"/>

</attribute>

</optional>

</define>

### Formatting Style

The style:data-style-name attribute refers to the data style used to format the numeric value. For general information about styles, see Chapter 14. For more information about data styles, see section 14.7.

For string variables this attribute must be omitted. Otherwise, this attribute is required.

The name must match the name of a data style.

This attribute can be used with any of the following elements:

* <text:date>
* <text:time>
* <text:page-number>
* <text:variable-set>
* <text:variable-get>
* <text:variable-input>
* <text:user-field-get>
* <text:user-field-input>
* <text:expression>

<define name="common-field-data-style-name-attlist">

<optional>

<attribute name="style:data-style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

</define>

### Number Formatting Style

Numbers that are used for number sequences such as page numbers or sequence fields can be formatted according to the number styles described in section 12.2. The number styles supported are as follows:

* Numeric: 1, 2, 3, ...
* Alphabetic: a, b, c, ... or A, B, C, ...
* Roman: i, ii, iii, iv, ... or I, II, III, IV,...

**Note**: The value of this attribute can be any of the [XSLT] number format keys 1, i, I, a, or A.

Alphabetic number styles need an additional attribute to determine how to display numbers that cannot be represented by a single letter. The OpenDocument format supports:

* Synchronized letter numbering, where letters are used multiple times, for example aa, bb, cc, and so on.
* Non-synchronized letter numbering, for example aa, ab, ac, and so on.

See section 12.2 for more information.

<define name="common-field-num-format-attlist">

<optional>

<ref name="common-num-format-attlist"/>

</optional>

</define>

1. Text Indices

OpenDocument text documents may contain automatically generated indices. An index generally contains a sorted list of all items of a certain types, where the sorting (document position, alphabetical, etc.) and the type of items (chapter headings, tables, etc.) are determined by the specific type of index.

## Index Marks

There are three types of index marks that correspond to the three types of index that make use of index marks. The three types of index marks are:

* Table of content index marks
* User-defined index marks
* Alphabetical index marks

The XML code for index marks is similar to the code for Bookmarks and References. The following are some basic rules about index marks:

* Each index mark is represented by a start and an end element.
* Both elements use an ID attribute to match the appropriate start and end elements.
* The start and end elements for an index mark must be contained in the same paragraph, with the start element occurring first.
* The attributes associated with the index mark are attached to the start element.
* The text between the start and end elements is the text the index entry.
* The formatting attributes for index marks can overlap.

### Table of Content Index Marks

The <text:toc-mark-start> element marks the start of a table of content index entry. The ID specified by the text:id attribute must be unique except for the matching index mark end element. There must be an end element to match the start element located in the same paragraph, with the start element appearing first.

<define name="paragraph-content" combine="choice">

<element name="text:toc-mark-start">

<ref name="text-toc-mark-start-attrs"/>

</element>

</define>

The attributes associated with the <text:toc-mark-start> element are:

* A text:id attribute to allow the start and end elements to be matched.
* A text:outline-level attribute to specify the outline level of the resulting table of content index entry.

<define name="text-toc-mark-start-attrs">

<ref name="text-id"/>

<ref name="text-outline-level"/>

</define>

<define name="text-outline-level">

<optional>

<attribute name="text:outline-level">

<ref name="positiveInteger"/>

</attribute>

</optional>

</define>

<define name="text-id">

<attribute name="text:id">

<ref name="string"/>

</attribute>

</define>

The <text:toc-mark-end> element marks the end of a table of contents index entry. There must be a start element with the same text:id value to match the end element located in the same paragraph, with the start element appearing first.

<define name="paragraph-content" combine="choice">

<element name="text:toc-mark-end">

<ref name="text-id"/>

</element>

</define>

Table of content index marks also have a variant that does not enclose the text to be indexed. This is represented using the <text:toc-mark> element which contains a text:string-value attribute for the text of the index entry. In this situation, a text:id attribute is not necessary because there are no start and end elements to match.

<define name="paragraph-content" combine="choice">

<element name="text:toc-mark">

<attribute name="text:string-value">

<ref name="string"/>

</attribute>

<ref name="text-outline-level"/>

</element>

</define>

### User-Defined Index Marks

The <text:user-index-mark-start> element marks the start of a user-defined index entry. The ID specified by the text:id attribute must be unique except for the matching index mark end element. There must be an end element to match the start element located in the same paragraph, with the start element appearing first.

<define name="paragraph-content" combine="choice">

<element name="text:user-index-mark-start">

<ref name="text-id"/>

<ref name="text-outline-level"/>

<ref name="text-index-name"/>

</element>

</define>

The <text:user-index-mark-end> element marks the end of the user-defined index entry. There must be a start element with the same text:id value to match the end element located in the same paragraph, with the start element appearing first.

<define name="paragraph-content" combine="choice">

<element name="text:user-index-mark-end">

<ref name="text-id"/>

<ref name="text-outline-level"/>

</element>

</define>

User index marks also have a variant that does not enclose the text to be indexed. This is represented by the <text:user-index-mark> element which contains a text:string-value attribute for the text of the index entry. In this situation, the text:id attribute is not necessary because there are no start and end elements to match.

<define name="paragraph-content" combine="choice">

<element name="text:user-index-mark">

<attribute name="text:string-value">

<ref name="string"/>

</attribute>

<ref name="text-outline-level"/>

<ref name="text-index-name"/>

</element>

</define>

#### Name of User Index

There can be more than one user-defined index. In this case, the user index must be named using the text:index-name attribute. This attribute determines to which user-defined index an index mark belongs. If no name is given, the default user-defined index is used.

<define name="text-index-name">

<attribute name="text:index-name">

<ref name="string"/>

</attribute>

</define>

### Alphabetical Index Mark

The <text:alpha-index-mark-start> element marks the start of an alphabetical index entry. There are two optional attributes that may contain keys for alphabetical entries, which allows structuring of entries. There is also a Boolean attribute that determines if this entry is intended to be the main entry, if there are several equal entries.

The ID specified by the text:id attribute must be unique except for the matching index mark end element. There must be an end element to match the start element located in the same paragraph, with the start element appearing first.

<define name="paragraph-content" combine="choice">

<element name="text:alphabetical-index-mark-start">

<ref name="text-id"/>

<ref name="text-alphabetical-index-mark-attrs"/>

</element>

</define>

The attributes associated with the <text:toc-mark-start> element are:

* A text:id attribute to allow the start and end elements to be matched.
* Additional keys
* Main entry

The <text:alpha-index-mark-end> element marks the end of an alphabetical index entry. There must be a start element with the same text:id value to match the end element located in the same paragraph, with the start element appearing first.

<define name="paragraph-content" combine="choice">

<element name="text:alphabetical-index-mark-end">

<ref name="text-id"/>

</element>

</define>

Alphabetical index marks also have a variant that does not enclose the text to be indexed. This is represented using the <text:alpha-index-mark> element which contains a text:string-value attribute for the text of the index entry. In this situation, a text:id attribute is not necessary because there are no start and end elements to match.

<define name="paragraph-content" combine="choice">

<element name="text:alphabetical-index-mark">

<attribute name="text:string-value">

<ref name="string"/>

</attribute>

<ref name="text-alphabetical-index-mark-attrs"/>

</element>

</define>

#### Additional Keys

The text:key1 and text:key2 attributes specify additional keys for the alphabetical index mark. If only one key is used, it must be contained in the text:key1 attribute.

<define name="text-alphabetical-index-mark-attrs" combine="interleave">

<optional>

<attribute name="text:key1">

<ref name="string"/>

</attribute>

</optional>

<optional>

<attribute name="text:key2">

<ref name="string"/>

</attribute>

</optional>

</define>

#### Phonetic Keys

For ideographic languages, there sometimes is no obvious or common sorting of the language's characters. One common scheme to facilitate an alphabetical index in such languages is to sort according to a phonetic description of the search time. To achieve this in the OpenDocument file format, there are additional attributes for the string value and the two keys for phonetic descriptions. The original value and key attributes are for display, but if phonetic variants are present, they should be used for sorting the index.

<define name="text-alphabetical-index-mark-attrs" combine="interleave">

<optional>

<attribute name="text:string-value-phonetic">

<ref name="string"/>

</attribute>

</optional>

<optional>

<attribute name="text:key1-phonetic">

<ref name="string"/>

</attribute>

</optional>

<optional>

<attribute name="text:key2-phonetic">

<ref name="string"/>

</attribute>

</optional>

</define>

#### Main Entry

If there are several index marks for the same entry, one of these entries may be declared as the main entry using the text:main-entry attribute.

<define name="text-alphabetical-index-mark-attrs" combine="interleave">

<optional>

<attribute name="text:main-entry" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

### Bibliography Index Mark

The <text:bibliography-mark> element contains the text and information for a bibliography index entry. It supports attributes for each type of bibliographical data that a bibliography index may contain.

<define name="paragraph-content" combine="choice">

<element name="text:bibliography-mark">

<attribute name="text:bibliography-type">

<ref name="text-bibliography-types"/>

</attribute>

<zeroOrMore>

<attribute>

<choice>

<name>text:identifier</name>

<name>text:address</name>

<name>text:annote</name>

<name>text:author</name>

<name>text:booktitle</name>

<name>text:chapter</name>

<name>text:edition</name>

<name>text:editor</name>

<name>text:howpublished</name>

<name>text:institution</name>

<name>text:journal</name>

<name>text:month</name>

<name>text:note</name>

<name>text:number</name>

<name>text:organizations</name>

<name>text:pages</name>

<name>text:publisher</name>

<name>text:school</name>

<name>text:series</name>

<name>text:title</name>

<name>text:report-type</name>

<name>text:volume</name>

<name>text:year</name>

<name>text:url</name>

<name>text:custom1</name>

<name>text:custom2</name>

<name>text:custom3</name>

<name>text:custom4</name>

<name>text:custom5</name>

<name>text:isbn</name>

<name>text:issn</name>

</choice>

<ref name="string"/>

</attribute>

</zeroOrMore>

<text/>

</element>

</define>

<define name="text-bibliography-types">

<choice>

<value>article</value>

<value>book</value>

<value>booklet</value>

<value>conference</value>

<value>custom1</value>

<value>custom2</value>

<value>custom3</value>

<value>custom4</value>

<value>custom5</value>

<value>email</value>

<value>inbook</value>

<value>incollection</value>

<value>inproceedings</value>

<value>journal</value>

<value>manual</value>

<value>mastersthesis</value>

<value>misc</value>

<value>phdthesis</value>

<value>proceedings</value>

<value>techreport</value>

<value>unpublished</value>

<value>www</value>

</choice>

</define>

## Index Structure

An index consists of two parts: The index source, and the index body. Both of these are contained in an element of their own, which in turn form the two child elements for the index element itself.

The index source is specific to the type of index it is being used for. It contains the information necessary to generate the index content. An index source has no graphical rendition.

The index body is the same for all types of indices. It contains the text generated from the information in the index source. The text contained in an index body is in no way special or different from text used elsewhere in this specification.

The content of the index body can be regenerated at any time from the information contained in the index source and the remainder of the document. One could say that the index source contains all the logical information about an index, while the index body contains the rendition of the index. A tool extracting structure information about a document might look only at the index source, while a rendering program might look only at an index body.

### Index Source

An index source element contains the information necessary to generate the index body. In addition to a set of flags that determine which information to include in an index, the index source contains a set of index templates. Such a template determines how an item to be contained in the index is to be rendered.

For example, a table of content might look as follows:

1 Introduction 7

1.1 Namespaces 7

1.2 Relax-NG Schema Prefix 8

An index source for this index would contain flags indicating that chapter headers at least up to level 2 are to be included. The contained index templates would define that an entry consists of the chapter number, a space, the chapter name, a tab (with a '.' leader) and the page number.

The various index templates are described together with their index elements. The index templates elements in use are described in section 7.12.

The different index source elements are described together with their corresponding index elements.

### Index Body Section

The index body contains the current textual rendition of the index. The format is the same as for regular text within this specification, e.g., text sections, except that it also allows index title sections.

<define name="text-index-body">

<element name="text:index-body">

<zeroOrMore>

<ref name="index-content-main"/>

</zeroOrMore>

</element>

</define>

<define name="index-content-main">

<choice>

<ref name="text-content"/>

<ref name="text-index-title"/>

</choice>

</define>

### Index Title Section

The index title is usually contained in a section of its own. The reason for this enclosure is to enable the popular layout of having an index title across the entire page, but having the index itself in a two column layout.

<define name="text-index-title">

<element name="text:index-title">

<ref name="sectionAttr"/>

<zeroOrMore>

<ref name="index-content-main"/>

</zeroOrMore>

</element>

</define>

## Table Of Content

A table of contents provides the user with a guide through the content of the document. It is typically found at the beginning of a document, contains the chapter headings with their respective page numbers. An example for a table of content may be found at the beginning of this document.

The items that can be listed in a table of content are:

* Headers (as defined by the outline structure of the document), up to a selectable level
* Table of content index marks
* Paragraphs formatted with a set of selectable paragraph styles

The table of contents is represented by the <text:table-of-content> element. The <text:table-of-content> element supports the same style (and class) attributes as a text section (see section 4.4).

<define name="text-table-of-content">

<element name="text:table-of-content">

<ref name="sectionAttr"/>

<ref name="text-table-of-content-source"/>

<ref name="text-index-body"/>

</element>

</define>

### Table of Content Source

The <text:table-of-content-source> element specifies how the table of contents is generated. It specifies how the entries are gathered.

The <text:table-of-content-source> element contains

* an optional template for the index title
* optional templates for index entries, one per level
* optionally a list of styles to be used for gathering index entries

<define name="text-table-of-content-source">

<element name="text:table-of-content-source">

<ref name="text-table-of-content-source-attlist"/>

<optional>

<ref name="text-index-title-template"/>

</optional>

<zeroOrMore>

<ref name="text-table-of-content-entry-template"/>

</zeroOrMore>

<zeroOrMore>

<ref name="text-index-source-styles"/>

</zeroOrMore>

</element>

</define>

The attributes that may be associated with the <text:table-of-content-source> element are:

* Outline level
* Use outline
* Use index marks
* Use index source styles
* Index source
* Relative tab stop position

#### Outline Level

The text:outline-level attribute specifies which outline levels are used when generating the table of contents.

The value of this attribute must be an integer greater than zero. If this attribute is omitted, all outline levels are used by default.

<define name="text-table-of-content-source-attlist" combine="interleave">

<optional>

<attribute name="text:outline-level">

<choice>

<ref name="positiveInteger"/>

</choice>

</attribute>

</optional>

</define>

#### Use Outline

The text:use-outline-level attribute determines whether headings are used to generate index entries. If the value is true, the table of contents includes entries generated from headings. The text:outline-level attribute specifies up to which level headings are being included. See section 7.1 for more information on index marks.

<define name="text-table-of-content-source-attlist" combine="interleave">

<optional>

<attribute name="text:use-outline-level" a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Use Index Marks

The text:use-index-marks attribute determines whether or not index marks are used to generate index entries. If the value is true, the table of contents includes entries generated from table of content index marks. The text:outline-level attribute specifies up to which level index marks are being included. See section 7.1 for more information on index marks.

<define name="text-table-of-content-source-attlist" combine="interleave">

<optional>

<attribute name="text:use-index-marks">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Use Index Source Styles

The text:use-index-source-styles attribute determines whether or not index entries are generated for paragraph formatted using certain paragraph styles. If the value is true, the table of contents includes an entry for every paragraph formatted with one of the styles specified in a <text:index-source-style> element. The text:outline-level attribute specifies up to which level index source styles are being included.

<define name="text-table-of-content-source-attlist" combine="interleave">

<optional>

<attribute name="text:use-index-source-styles">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Index Scope

The text:index-scope attribute determines whether the table-of-content is generated for the whole document, or only for the current chapter.

<define name="text-table-of-content-source-attlist" combine="interleave">

<optional>

<attribute name="text:index-scope">

<choice>

<value>document</value>

<value>chapter</value>

</choice>

</attribute>

</optional>

</define>

#### Relative Tab-Stop Position

The text:relative-tab-stop-position attribute determines whether the position of tab stops is relative to the left margin or to the left indent as determined by the paragraph style. This is useful for copying the same entry configuration for all outline levels because with relative tab stop positions the tabs do not need to be adjusted to the respective paragraph format.

<define name="text-table-of-content-source-attlist" combine="interleave">

<optional>

<attribute name="text:relative-tab-stop-position">

<ref name="boolean"/>

</attribute>

</optional>

</define>

### Table of Content Entry Template

The <text:table-of-content-entry-template> element determines the format of an index entry for a particular outline level. For each table of content, there must not be more than one element for any outline level. (See below.)

<define name="text-table-of-content-entry-template">

<element name="text:table-of-content-entry-template">

<ref name="text-table-of-content-entry-template-attlist"/>

<zeroOrMore>

<ref name="text-table-of-content-children"/>

</zeroOrMore>

</element>

</define>

A table of content entry template supports the following kinds of text elements:

* Chapter and Page Number
* Reference Text
* Text Span
* Tab
* Hyperlink start and end

<define name="text-table-of-content-children">

<choice>

<ref name="text-index-entry-chapter"/>

<ref name="text-index-entry-page-number"/>

<ref name="text-index-entry-text"/>

<ref name="text-index-entry-span"/>

<ref name="text-index-entry-tab-stop"/>

<ref name="text-index-entry-link-start"/>

<ref name="text-index-entry-link-end"/>

</choice>

</define>

The attributes that may be associated associate with the <text:table-of-content-entry-template> element are:

* Template outline level
* Paragraph style

#### Template Outline Level

This attribute specifies to which outline level the entry configuration applies. Outline levels must be unique for the template elements in one index source.

<define name="text-table-of-content-entry-template-attlist"

combine="interleave">

<attribute name="text:outline-level">

<ref name="positiveInteger"/>

</attribute>

</define>

#### Paragraph Style

The text:style-name attribute specifies the paragraph style to use for this template.

<define name="text-table-of-content-entry-template-attlist"

combine="interleave">

<attribute name="text:style-name">

<ref name="styleNameRef"/>

</attribute>

</define>

## Index of Illustrations

The index of illustrations lists all images and graphics in the current document or chapter. The index entries can be derived from the caption of the illustration or the name of the illustration.

The attribute that may be attached to the <text:illustration-index> element is:

* text:style-name

This attribute specifies the section style to use for the index of illustrations.

<define name="text-illustration-index">

<element name="text:illustration-index">

<ref name="sectionAttr"/>

<ref name="text-illustration-index-source"/>

<ref name="text-index-body"/>

</element>

</define>

### Index of Illustration Source

The <text:illustration-index-source> element specifies how the index of illustrations is generated.

<define name="text-illustration-index-source">

<element name="text:illustration-index-source">

<ref name="text-illustration-index-source-attrs"/>

<optional>

<ref name="text-index-title-template"/>

</optional>

<optional>

<ref name="text-illustration-index-entry-template"/>

</optional>

</element>

</define>

The attributes that may be associated with a <text:illustration-index-source> element are:

* Use caption
* Caption sequence name
* Caption sequence format
* Index scope

This attribute specifies whether the index applies to the entire document or only the the current chapter.

* text:relative-tab-stop-position

This attribute specifies whether the position of tab stops are interpreted relative to the left margin or the left indent.

<define name="text-illustration-index-source-attrs" combine="interleave">

<ref name="text-index-scope-attr"/>

</define>

<define name="text-index-scope-attr">

<optional>

<attribute name="text:index-scope" a:defaultValue="document">

<choice>

<value>document</value>

<value>chapter</value>

</choice>

</attribute>

</optional>

</define>

<define name="text-illustration-index-source-attrs" combine="interleave">

<ref name="text-relative-tab-stop-position-attr"/>

</define>

<define name="text-relative-tab-stop-position-attr">

<optional>

<attribute name="text:relative-tab-stop-position"

a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Use Caption

Each object contained in a text document has a name. In addition, images also have a caption. The image caption or the image name can be gathered for the index of illustrations.

<define name="text-illustration-index-source-attrs" combine="interleave">

<optional>

<attribute name="text:use-caption" a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Caption Sequence Name

Captions are associated with a sequence name. If the text:use-caption attribute is set to true, this attribute must be used to specify the sequence with which the captions are associated.

If this attribute is omitted, the default sequence for the object type is used, for example the sequence “Illustration” is used for illustrations.

<define name="text-illustration-index-source-attrs" combine="interleave">

<optional>

<attribute name="text:caption-sequence-name">

<ref name="string"/>

</attribute>

</optional>

</define>

#### Caption Sequence Format

If the entries for the index of illustrations are obtained from the image captions, this attribute must be used to specify the format for the entries.

<define name="text-illustration-index-source-attrs" combine="interleave">

<optional>

<attribute name="text:caption-sequence-format">

<choice>

<value>text</value>

<value>category-and-value</value>

<value>caption</value>

</choice>

</attribute>

</optional>

</define>

### Illustration Index Entry Template

The illustration index entry template element determines the format of an index entry for a particular outline level.

<define name="text-illustration-index-entry-template">

<element name="text:illustration-index-entry-template">

<ref name="text-illustration-index-entry-content"/>

</element>

</define>

<define name="text-illustration-index-entry-content">

<ref name="text-illustration-index-entry-template-attrs"/>

<zeroOrMore>

<choice>

<ref name="text-index-entry-page-number"/>

<ref name="text-index-entry-text"/>

<ref name="text-index-entry-span"/>

<ref name="text-index-entry-tab-stop"/>

</choice>

</zeroOrMore>

</define>

The attribute that may be associated with the <text:illustration-index-entry-template> element is:

* Paragraph style

#### Paragraph Style

This attribute identifies the paragraph style to use for this template.

<define name="text-illustration-index-entry-template-attrs">

<attribute name="text:style-name">

<ref name="styleNameRef"/>

</attribute>

</define>

## Index of Tables

The index of tables lists all of the tables in the current document or chapter. It works in exactly the same way as the index of illustrations.

<define name="text-table-index">

<element name="text:table-index">

<ref name="sectionAttr"/>

<ref name="text-table-index-source"/>

<ref name="text-index-body"/>

</element>

</define>

### Table Index Source

The <text:table-index-source> element specifies how the index of tables is generated.

The attributes that may be associated with this element are the same as those that can be associated with the <text:illustration-index-source> element. See section 7.4.1 for detailed information about these attributes.

<define name="text-table-index-source">

<element name="text:table-index-source">

<ref name="text-illustration-index-source-attrs"/>

<optional>

<ref name="text-index-title-template"/>

</optional>

<optional>

<ref name="text-table-index-entry-template"/>

</optional>

</element>

</define>

### Table Index Entry Template

The table index entry template element determines the format of an index entry for a particular outline level.

The attributes that may be associated with this element are the same as those that can be associated with the <text:illustration-index-entry-template> element. See section 7.4.2 for detailed information about these attributes.

<define name="text-table-index-entry-template">

<element name="text:table-index-entry-template">

<ref name="text-illustration-index-entry-content"/>

</element>

</define>

## Index of Objects

The index of objects lists all of the objects in the current document or chapter. It gathers its entries from the known object types.

<define name="text-object-index">

<element name="text:object-index">

<ref name="sectionAttr"/>

<ref name="text-object-index-source"/>

<ref name="text-index-body"/>

</element>

</define>

### Object Index Source

The <text:object-index-source> element determines which object types to include in the index of objects. It also supports the standard index source attributes.

<define name="text-object-index-source">

<element name="text:object-index-source">

<ref name="text-object-index-source-attrs"/>

<optional>

<ref name="text-index-title-template"/>

</optional>

<optional>

<ref name="text-object-index-entry-template"/>

</optional>

</element>

</define>

The attributes that may be associated with the <text:object-index-source> element are:

* Use attributes, text:use-\*-objects
* Index scope (see section 7.4.1)

This attribute specifies whether the index applies to the entire document or only the the current chapter.

* Relative tab stop position (see section 7.4.1)

This attribute specifies whether the position of tab stops are interpreted relative to the left margin or the left indent.

<define name="text-object-index-source-attrs" combine="interleave">

<ref name="text-index-scope-attr"/>

</define>

<define name="text-object-index-source-attrs" combine="interleave">

<ref name="text-relative-tab-stop-position-attr"/>

</define>

#### Use Attributes

The text:use-\*-objects attributes specify which types of objects to include in the index of objects. There is an attribute for each type of object as follows:

* text:use-spreadsheet-objects
* text:use-draw-objects
* text:use-chart-objects
* text:use-math-objects

Other objects are included or omitted using the following attribute:

* text:use-other-objects

<define name="text-object-index-source-attrs" combine="interleave">

<optional>

<attribute name="text:use-spreadsheet-objects" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

<define name="text-object-index-source-attrs" combine="interleave">

<optional>

<attribute name="text:use-math-objects" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

<define name="text-object-index-source-attrs" combine="interleave">

<optional>

<attribute name="text:use-draw-objects" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

<define name="text-object-index-source-attrs" combine="interleave">

<optional>

<attribute name="text:use-chart-objects" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

<define name="text-object-index-source-attrs" combine="interleave">

<optional>

<attribute name="text:use-other-objects" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

### Object Index Entry Template

The object index entry template element determines the format of an index entry for a particular outline level.

<define name="text-object-index-entry-template">

<element name="text:object-index-entry-template">

<ref name="text-illustration-index-entry-content"/>

</element>

</define>

The attributes that may be associated with this element are the same as those that can be associated with the <text:illustration-index-entry-template> element. See section 7.4.2 for detailed information about these attributes.

## User-Defined Index

A user-defined index combines the capabilities of the indexes discussed earlier in this chapter. A user-defined index can gather entries from the following sources:

* Index marks
* Paragraphs formatted using particular paragraph styles
* Tables, images, or objects
* Text frames

The <text:user-index> element represents a user-defined index.

<define name="text-user-index">

<element name="text:user-index">

<ref name="sectionAttr"/>

<ref name="text-user-index-source"/>

<ref name="text-index-body"/>

</element>

</define>

### User-Defined Index Source

The <text:user-index-source> element can contain several attributes that determine how the index entries are gathered. It also supports an attribute that determines how the outline levels of the index entries are gathered.

The paragraph formats that are used as index marks are encoded in <text:index-source-styles> elements, just like in <text:table-of-content-source> elements.

<define name="text-user-index-source">

<element name="text:user-index-source">

<ref name="text-user-index-source-attr"/>

<optional>

<ref name="text-index-title-template"/>

</optional>

<zeroOrMore>

<ref name="text-user-index-entry-template"/>

</zeroOrMore>

<zeroOrMore>

<ref name="text-index-source-styles"/>

</zeroOrMore>

</element>

</define>

The attributes that may be associated with <text:user-index-source> elements are:

* Use attributes, text:use-\*
* Copy outline level
* Index scope (see section 7.4.1)

This attribute specifies whether the index applies to the entire document or only to the current chapter.

* Index name

In order to support several user-defined indexes with different contents, user index marks have a text:index-name attribute. The same attribute can be used with a <text:user-index-source> element to specify which index marks apply to the current index.

* Relative tab stop position (see section 7.4.1)

This attribute specifies whether the position of tab stops are interpreted relative to the left margin or the left indent.

<define name="text-user-index-source-attr" combine="interleave">

<ref name="text-index-scope-attr"/>

<ref name="text-relative-tab-stop-position-attr"/>

<attribute name="text:index-name">

<ref name="string"/>

</attribute>

</define>

#### Use Attributes

The text:use-\* attributes specify which entries to include in the user-defined index. The following attributes exist:

* text:use-index-marks
* text:use-graphics
* text:use-tables
* text:use-floating-frames
* text:use-objects

<define name="text-user-index-source-attr" combine="interleave">

<optional>

<attribute name="text:use-index-marks" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

<optional>

<attribute name="text:use-graphics" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

<optional>

<attribute name="text:use-tables" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

<optional>

<attribute name="text:use-floating-frames"

a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

<optional>

<attribute name="text:use-objects" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Copy Outline Levels

This attribute can have a value of true or false.

If the value is true, the entries are gathered at the outline level of the source element to which they refer.

If the value is false, all index entries gathered are at the top outline level. For example, if an image appears in section 1.2.3, the entry for the image is located at outline level 3.

<define name="text-user-index-source-attr" combine="interleave">

<optional>

<attribute name="text:copy-outline-levels"

a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

### User-Defined Index Entry Template

User index entry templates support entry elements for chapter number, page number, entry text, text spans, and tab stops.

<define name="text-user-index-entry-template">

<element name="text:user-index-entry-template">

<ref name="text-user-index-entry-template-attrs"/>

<zeroOrMore>

<choice>

<ref name="text-index-entry-chapter"/>

<ref name="text-index-entry-page-number"/>

<ref name="text-index-entry-text"/>

<ref name="text-index-entry-span"/>

<ref name="text-index-entry-tab-stop"/>

</choice>

</zeroOrMore>

</element>

</define>

The attributes that may be associated with the <text:user-index-entry-template> elements are:

* Template outline level
* Paragraph style

#### Template Outline Level

The text:outline-level attribute specifies to which outline level this entry configuration applies.

All <text:outline-level> elements that are contained in the same parent element must specify different outline levels.

<define name="text-user-index-entry-template-attrs" combine="interleave">

<attribute name="text:outline-level">

<ref name="positiveInteger"/>

</attribute>

</define>

#### Paragraph Style

The text:style-name attribute specifies the paragraph style to use for the template.

<define name="text-user-index-entry-template-attrs" combine="interleave">

<attribute name="text:style-name">

<ref name="styleNameRef"/>

</attribute>

</define>

## Alphabetical Index

An alphabetical index gathers its entries solely from index marks.

<define name="text-alphabetical-index">

<element name="text:alphabetical-index">

<ref name="sectionAttr"/>

<ref name="text-alphabetical-index-source"/>

<ref name="text-index-body"/>

</element>

</define>

### Alphabetical Index Source

The <text:alphabetical-index-source> element specifies how the alphabetical index is generated.

<define name="text-alphabetical-index-source">

<element name="text:alphabetical-index-source">

<ref name="text-alphabetical-index-source-attrs"/>

<optional>

<ref name="text-index-title-template"/>

</optional>

<zeroOrMore>

<ref name="text-alphabetical-index-entry-template"/>

</zeroOrMore>

</element>

</define>

The attributes that may be associated with <text:alphabetical-index-source> elements are:

* Ignore case
* Main entry style name
* Alphabetical separators
* Combine entries attributes
* Use keys as entries
* Capitalize entries
* Comma separated entries
* Sort language, country and algorithm
* Index scope (see section 7.4.1)

This attribute specifies whether the index applies to the entire document or only to the current chapter.

* Relative tab stop position (see section 7.4.1)

This attribute specifies whether the position of tab stops are interpreted relative to the left margin or the left indent.

<define name="text-alphabetical-index-source-attrs" combine="interleave">

<ref name="text-index-scope-attr"/>

<ref name="text-relative-tab-stop-position-attr"/>

</define>

#### Ignore Case

The text:ignore-case attribute determines whether or not the capitalization of words is ignored. If the value is true, the capitalization is ignored and entries that are identical except for character case are listed as the same entries. If the value is false, the capitalization of words is not ignored.

<define name="text-alphabetical-index-source-attrs" combine="interleave">

<optional>

<attribute name="text:ignore-case" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Main Entry Style Name

The text:main-entry-style-name attribute determines the character style to use for main entries. Sub entries are formatted using the default character style determined by the paragraph style of the entries.

<define name="text-alphabetical-index-source-attrs" combine="interleave">

<optional>

<attribute name="text:main-entry-style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

</define>

#### Alphabetical Separators

The text:alphabetical-separators attribute determines whether or not entries beginning with the same letter are grouped and separated from the entries beginning with the next letter, and so on.

The value of this attribute can be true or false.

If the value is true, all entries beginning with the same letter are grouped together. The index contains headings for each section, for example, A for all entries starting with the letter A, B for all entries starting with the letter B, and so on.

<define name="text-alphabetical-index-source-attrs" combine="interleave">

<optional>

<attribute name="text:alphabetical-separators" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Combining Entries

There are several options for dealing with the common situation where there are multiple index entries for the same word or phrase, as follows:

* Multiple entries for the same word can be combined into a single entry using the text:combine-entries attribute.
* The pages referenced by a combined entry can be formatted as:
  + As a range of numbers separated by a dash using the text:combine-entries-with-dash attribute
  + As the start number with a pp label, or the appropriate label for the chosen language, using the text:combine-entries-with-pp attribute

<define name="text-alphabetical-index-source-attrs" combine="interleave">

<optional>

<attribute name="text:combine-entries" a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

<optional>

<attribute name="text:combine-entries-with-dash"

a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

<optional>

<attribute name="text:combine-entries-with-pp" a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

**Example**: Combining index entries

An index mark for the word “XML” occurs on pages 45, 46, 47, and 48. The entries can be formatted as follows:

|  |  |
| --- | --- |
| **Entry formatted as** | **Result** |
| Separate entries | XML 45 XML 46 etc. |
| Simple combined entries | XML 45, 46, 47, 48 |
| Entries combined with dash | XML 45-48 |
| Entries combined with pp | XML 45pp |

#### Use Keys as Entries

In addition to a keyword, index marks can have up to two keys. If the value of this attribute is true, the keys are used as additional entries. If the value of this attribute is false, the keys are used as sub entries.

<define name="text-alphabetical-index-source-attrs" combine="interleave">

<optional>

<attribute name="text:use-keys-as-entries" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Capitalize Entries

The text:capitalize-entries attribute determines whether or not the entries in the index are to be capitalized.

<define name="text-alphabetical-index-source-attrs" combine="interleave">

<optional>

<attribute name="text:capitalize-entries" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Comma Separated Entries

The text:comma-separated attribute specifies how to treat multiple index entries. Instead of listing each index entry on a separate line, multiple entries can be listed on a single line separated by a comma. If the value of this attribute is true, multiple entries are listed on a single line separated by a comma. By default, the value of this attribute is false and each index entry is displayed on a separate line.

<define name="text-alphabetical-index-source-attrs" combine="interleave">

<optional>

<attribute name="text:comma-separated" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Sort country, Language, and Algorithm

If index entries are to be sorted, these attributes can be used to specify the sorting. The attributes country and language specify the sorting locale. For some locales, there are multiple sorting algorithms in use. In this case, the algorithm attribute can be used to specify an algorithm by name.

<define name="text-alphabetical-index-source-attrs" combine="interleave">

<optional>

<attribute name="fo:language">

<ref name="languageCode"/>

</attribute>

</optional>

</define>

<define name="text-alphabetical-index-source-attrs" combine="interleave">

<optional>

<attribute name="fo:country">

<ref name="countryCode"/>

</attribute>

</optional>

</define>

<define name="text-alphabetical-index-source-attrs" combine="interleave">

<optional>

<attribute name="text:sort-algorithm">

<ref name="string"/>

</attribute>

</optional>

</define>

### Auto Mark File

The alphabetical index supports a so-called auto mark file. Such a file contains a list of terms, and each occurrence of such a term is to be included in the alphabetical index. The alphabetical index mark file is declared as part of the text declarations (see section 4.8). The declaration element in an XLink, which points to the resource containing the list of terms.

<define name="text-alphabetical-index-auto-mark-file">

<element name="text:alphabetical-index-auto-mark-file">

<attribute name="xlink:href">

<ref name="anyURI"/>

</attribute>

<optional>

<attribute name="xlink:type" a:defaultValue="simple">

<value>simple</value>

</attribute>

</optional>

</element>

</define>

### Alphabetical Index Entry Template

Alphabetical indexes support three levels; one level for the main index entry, and up to two additional levels for keys associated with the index entries. Alphabetical indexes also use an entry template for the alphabetical separator.

<define name="text-alphabetical-index-entry-template">

<element name="text:alphabetical-index-entry-template">

<ref name="text-alphabetical-index-entry-template-attrs"/>

<zeroOrMore>

<choice>

<ref name="text-index-entry-chapter"/>

<ref name="text-index-entry-page-number"/>

<ref name="text-index-entry-text"/>

<ref name="text-index-entry-span"/>

<ref name="text-index-entry-tab-stop"/>

</choice>

</zeroOrMore>

</element>

</define>

The attributes that may be associated with the <text:alphabetical-index-entry-template> elements are:

* Template outline level
* Paragraph style

#### Template Outline Level

This attribute specifies whether the template applies to:

* One of the three levels 1,2,or 3

or

* The alphabetical separator

<define name="text-alphabetical-index-entry-template-attrs"

combine="interleave">

<attribute name="text:outline-level">

<choice>

<value>1</value>

<value>2</value>

<value>3</value>

<value>separator</value>

</choice>

</attribute>

</define>

#### Paragraph Style

The text:style-name attribute specifies the paragraph style to use for the template.

<define name="text-alphabetical-index-entry-template-attrs"

combine="interleave">

<attribute name="text:style-name">

<ref name="styleNameRef"/>

</attribute>

</define>

## Bibliography

A bibliography index gathers its entries from bibliography index marks. The <text:bibliography> element represents a bibliography.

<define name="text-bibliography">

<element name="text:bibliography">

<ref name="sectionAttr"/>

<ref name="text-bibliography-source"/>

<ref name="text-index-body"/>

</element>

</define>

### Bibliography Index Source

The <text:bibliography-source> element specifies how the bibliography is generated.

<define name="text-bibliography-source">

<element name="text:bibliography-source">

<optional>

<ref name="text-index-title-template"/>

</optional>

<zeroOrMore>

<ref name="text-bibliography-entry-template"/>

</zeroOrMore>

</element>

</define>

### Bibliography Entry Template

Bibliography entry templates support entry elements for bibliography data, text spans, and tab stops. There is one entry template element for each type of entry.

<define name="text-bibliography-entry-template">

<element name="text:bibliography-entry-template">

<ref name="text-bibliography-entry-template-attrs"/>

<zeroOrMore>

<choice>

<ref name="text-index-entry-span"/>

<ref name="text-index-entry-tab-stop"/>

<ref name="text-index-entry-bibliography"/>

</choice>

</zeroOrMore>

</element>

</define>

The attributes that may be associated with the <text:bibliography-entry-template> elements are:

* Bibliography type
* Paragraph style

#### Bibliography Type

This attribute specifies to which type of bibliographical entry the template applies. This attribute must be unique among all <text:bibliography-type> elements within the same parent element.

<define name="text-bibliography-entry-template-attrs" combine="interleave">

<attribute name="text:bibliography-type">

<ref name="text-bibliography-types"/>

</attribute>

</define>

#### Paragraph Style

The text:style-name attribute specifies the paragraph style to use for this template.

<define name="text-bibliography-entry-template-attrs" combine="interleave">

<attribute name="text:style-name">

<ref name="styleNameRef"/>

</attribute>

</define>

## index source styles

Some indices can gather index entries from paragraphs formatted using certain paragraph styles. The <text:index-source-styles> element contains all of the <text:index-source-style> elements for a particular outline level. The text:outline-levels attribute determines at which outline level to list the index entries gathered from the respective paragraph styles. There can only be one <text:index-source-style> element for each outline level.

<define name="text-index-source-styles">

<element name="text:index-source-styles">

<attribute name="text:outline-level">

<ref name="positiveInteger"/>

</attribute>

<zeroOrMore>

<ref name="text-index-source-style"/>

</zeroOrMore>

</element>

</define>

### Index source style

All paragraphs formatted using the style or class specified in the <text:index-source-style> element are included in the index.

<define name="text-index-source-style">

<element name="text:index-source-style">

<attribute name="text:style-name">

<ref name="styleName"/>

</attribute>

<empty/>

</element>

</define>

## Index title template

The <text:index-title-template> element determines the style and content of the index title. There can only be one <text:index-title-template> element contained in a <text:table-of-content-source> element.

<define name="text-index-title-template">

<element name="text:index-title-template">

<optional>

<attribute name="text:style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

<text/>

</element>

</define>

## Index Template Entries

There are eight types of index entries, as follows:

* Chapter information
* Entry text
* Page number
* Fixed string
* Bibliography information
* Tab stop
* Hyperlink start and end

### Chapter Information

The <text:index-entry-chapter> element displays the chapter number of the index entry. The character style for the chapter number can be included in the index entry element as a text:style-name attribute.

<define name="text-index-entry-chapter">

<element name="text:index-entry-chapter">

<optional>

<attribute name="text:style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

<ref name="text-index-entry-chapter-attrs"/>

</element>

</define>

**Note**: This element can only display the chapter number. To display the chapter name, the <text:index-entry-text> elements must be used.

#### Display Chapter Format

The text:display attribute displays either the chapter number, the chapter name, or both.

<define name="text-index-entry-chapter-attrs">

<optional>

<attribute name="text:display" a:defaultValue="number">

<choice>

<value>name</value>

<value>number</value>

<value>number-and-name</value>

</choice>

</attribute>

</optional>

</define>

### Entry Text

The <text:index-entry-text> element displays the text of the index entry, for example, the chapter name if the entry is derived from a header or the phrase contained in the index mark if the entry is derived from an index mark. The character style for the entry text can be included in the index entry element as a text:style-name attribute.

<define name="text-index-entry-text">

<element name="text:index-entry-text">

<optional>

<attribute name="text:style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

</element>

</define>

### Page Number

The <text:index-entry-page-number> element displays the page number on which the index entry is located. The character style for the page number can be included in the index entry element as a text:style-name attribute.

<define name="text-index-entry-page-number">

<element name="text:index-entry-page-number">

<optional>

<attribute name="text:style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

</element>

</define>

### Fixed String

The <text:index-entry-span> element represents a fixed string within an index entry. The character style for the entry text can be included in the index entry element as a text:style-name attribute. Unlike the <text:span> element, the <text:index-entry-span> element does not have any child elements.

<define name="text-index-entry-span">

<element name="text:index-entry-span">

<optional>

<attribute name="text:style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

<text/>

</element>

</define>

### Bibliography Information

The <text:index-entry-bibliography> element introduces bibliography data into index entry templates.

<define name="text-index-entry-bibliography">

<element name="text:index-entry-bibliography">

<ref name="text-index-entry-bibliography-attrs"/>

</element>

</define>

The attributes that may be associated with the <text:index-entry-bibliography> element are:

* text:style-name attribute
* text:bibliography-data-field attribute

#### Text Style Name

The text:style-name attribute determines the style for display of the entry.

<define name="text-index-entry-bibliography-attrs" combine="interleave">

<optional>

<attribute name="text:style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

</define>

#### Bibliography Data Field Identifier

The text:bibliography-data-field attribute determines which part of the bibliography data field will be displayed.

<define name="text-index-entry-bibliography-attrs" combine="interleave">

<attribute name="text:bibliography-data-field">

<choice>

<value>address</value>

<value>annote</value>

<value>author</value>

<value>bibliography-type</value>

<value>booktitle</value>

<value>chapter</value>

<value>custom1</value>

<value>custom2</value>

<value>custom3</value>

<value>custom4</value>

<value>custom5</value>

<value>edition</value>

<value>editor</value>

<value>howpublished</value>

<value>identifier</value>

<value>institution</value>

<value>isbn</value>

<value>issn</value>

<value>journal</value>

<value>month</value>

<value>note</value>

<value>number</value>

<value>organizations</value>

<value>pages</value>

<value>publisher</value>

<value>report-type</value>

<value>school</value>

<value>series</value>

<value>title</value>

<value>url</value>

<value>volume</value>

<value>year</value>

</choice>

</attribute>

</define>

### Tab Stop

The <text:index-entry-tab-stop> element represents a tab stop within an index entry. It also contains the position information for the tab stop.

<define name="text-index-entry-tab-stop">

<element name="text:index-entry-tab-stop">

<optional>

<attribute name="text:style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

<ref name="text-index-entry-tab-stop-attrs"/>

</element>

</define>

The attributes that may be associated with the <text:index-entry-tab-stop> element are:

* style:leader-char
* style:type
* style:position

#### Leader Char

The style:leader-char attribute specifies the leader character.

<define name="text-index-entry-tab-stop-attrs" combine="interleave">

<optional>

<attribute name="style:leader-char">

<ref name="character"/>

</attribute>

</optional>

</define>

#### Tab Type and Position

The style:type attribute specifies the tab stop type. The <text:index-entry-tab-stop> element only supports two types of tab: left and right.

If the value of this attribute is left, the style:position attribute must also be used. Otherwise, this attribute must be omitted. The style:position attribute specifies the position of the tab. Depending on the value of the text:relative-tab-stop-position attribute in the <text:index-entry-config> element, the position of the tab is interpreted as being relative to the left margin or the left indent.

<define name="text-index-entry-tab-stop-attrs" combine="interleave">

<choice>

<attribute name="style:type">

<value>right</value>

</attribute>

<group>

<attribute name="style:type">

<value>left</value>

</attribute>

<attribute name="style:position">

<ref name="length"/>

</attribute>

</group>

</choice>

</define>

### Hyperlink Start and End

The <text:index-entry-link-start> and <text:index-entry-link-end> elements mark the start and end of a hyperlink index entry. The character style for the hyperlink can be included in the index entry element as a text:style-name attribute.

<define name="text-index-entry-link-start">

<element name="text:index-entry-link-start">

<optional>

<attribute name="text:style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

</element>

</define>

<define name="text-index-entry-link-end">

<element name="text:index-entry-link-end">

<optional>

<attribute name="text:style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

</element>

</define>

### Example of an Index Entry Configuration

The following is an example of the XML code for a table of contents called Table of Content with the following characteristics:

* It uses the top two outline levels.
* Each entry consists of the chapter number, a closing parenthesis, the chapter title, a tab stop, and the page number.
* For the top outline level, the page number is formatted using a style called Bold.
* For the second outline level, a bracket is used instead of a closing parenthesis.

Example: Table of Content

<text:table-of-content>

<text:table-of-content-source

text:outline-level="2"

text:use-index-marks="false"

text:index-scope="document">

<text:index-title-template text:style-name="Index 1">

Table of Content

</text:index-title-template>

<text:index-entry-template

text:outline-level="1"

text:style-name="Contents 1">

<text:index-entry-chapter text:display="number"/>

<text:index-entry-span>) </text:index-entry-span>

<text:index-entry-text/>

<text:index-entry-tab-stop style:type="right"/>

<text:index-entry-page-number text:style-name="bold"/>

</text:index-entry-template>

<text:index-entry-template

text:outline-level="2"

text:style-name="Contents 2">

<text:index-entry-chapter text:display="number"/>

<text:index-entry-span>] </text:index-entry-span>

<text:index-entry-text/>

<text:index-entry-tab-stop style:type="right"/>

<text:index-entry-page-number/>

</text:index-entry-template>

</text:table-of-content-source>

<text:table-of-content-body>

*[... header ...]*

<text:p text:style-name="*[...]*">1) Chapter

<text:tab-stop/><text:span stylename="bold"> 1 </text:span>

</text:p>

<text:p text:style-name="*[...]*">1.1] Subchapter

<text:tab-stop/>1

</text:p>

*[... more entries ...]*

</text:table-of-content-body>

</text:table-of-content>

1. Tables

This chapter describes the table structure that is used for tables that are embedded within text documents and for spreadsheets.

## Basic Table Model

The structure of OpenDocument tables is similar to the structure of [HTML4] or [XSL] tables, and like these tables, they can be nested.

The representation of tables is based on a grid of rows and columns. Rows take precedence over columns. The table is divided into rows and the rows are divided into cells. Each column includes a column description, but this description does not contain any cells.

Table rows may be empty, and different rows might contain a different number of table cells. This is not an error, but applications might resolve this in different ways. Spreadsheet applications typically operate on large tables that have a fixed application dependent row and column number, but may have an unused area. Only the used area of the table is saved in files. When loading a table with empty or incomplete rows into a spreadsheet application, empty rows typically introduce a default row (just as in an empty sheet), and incomplete rows are filled with empty cells (just like in an empty sheet).

All other applications typically have fixed size tables. Incomplete rows are basically rendered as if they had the necessary number of empty cells, and the same applies to empty rows. Empty cells typically occupy the space of an empty paragraph.

Rows and columns appear in **row groups** and **column groups**. These groups specify whether or not to repeat a row or column on the next page.

### Table Element

The table element is the root element for tables.

<define name="table-table">

<element name="table:table">

<ref name="table-table-attlist"/>

<optional>

<ref name="table-table-source"/>

</optional>

<optional>

<ref name="office-dde-source"/>

</optional>

<optional>

<ref name="table-scenario"/>

</optional>

<optional>

<ref name="office-forms"/>

</optional>

<optional>

<ref name="table-shapes"/>

</optional>

<ref name="table-columns-and-groups"/>

<ref name="table-rows-and-groups"/>

</element>

</define>

The content models for tables is rather complex. The details are explained in the section 8.2. For the moment, it can be assumed that table element's content are columns and row elements.

<define name="table-columns-and-groups">

<oneOrMore>

<choice>

<ref name="table-table-column-group"/>

<ref name="table-columns-no-group"/>

</choice>

</oneOrMore>

</define>

<define name="table-columns-no-group">

<choice>

<group>

<ref name="table-columns"/>

<optional>

<ref name="table-table-header-columns"/>

<optional>

<ref name="table-columns"/>

</optional>

</optional>

</group>

<group>

<ref name="table-table-header-columns"/>

<optional>

<ref name="table-columns"/>

</optional>

</group>

</choice>

</define>

<define name="table-columns">

<choice>

<ref name="table-table-columns"/>

<oneOrMore>

<ref name="table-table-column"/>

</oneOrMore>

</choice>

</define>

<define name="table-rows-and-groups">

<oneOrMore>

<choice>

<ref name="table-table-row-group"/>

<ref name="table-rows-no-group"/>

</choice>

</oneOrMore>

</define>

<define name="table-rows-no-group">

<choice>

<group>

<ref name="table-rows"/>

<optional>

<ref name="table-table-header-rows"/>

<optional>

<ref name="table-rows"/>

</optional>

</optional>

</group>

<group>

<ref name="table-table-header-rows"/>

<optional>

<ref name="table-rows"/>

</optional>

</group>

</choice>

</define>

<define name="table-rows">

<choice>

<ref name="table-table-rows"/>

<oneOrMore>

<optional>

<ref name="text-soft-page-break"/>

</optional>

<ref name="table-table-row"/>

</oneOrMore>

</choice>

</define>

#### Table Name

The table:name attribute specifies the name of a table.

<define name="table-table-attlist" combine="interleave">

<optional>

<attribute name="table:name">

<ref name="string"/>

</attribute>

</optional>

</define>

#### Table Style

The table:style-name attribute references a table style, i.e., an <style:style> element of type “table”. The table style describes the formatting properties of the table, such as width and background color. The table style can be either an automatic or common style.

<define name="table-table-attlist" combine="interleave">

<optional>

<attribute name="table:style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

</define>

**Example: Table Style**

<style:style style:name="Table 1" style:family="table">

<style:table-properties style:width="12cm"

fo:background-color="light-grey"/>

</style:style>

<table:table table:name="Table 1" table:style-name="Table 1">

...

</table:table>

#### Protected

The table:protected attribute specifies whether or not a table is protected from editing. If the table is protected, the table:protection-key attribute can specify a password to prevent a user from resetting the protection flag to enable editing. If a table is protected, all of the table elements and the cell elements with a style:cell-protect attribute set to true are protected.

To avoid saving the password directly into the XML file, only a hash value of the password is stored within the table:protection-key attribute.

<define name="table-table-attlist" combine="interleave">

<optional>

<attribute name="table:protected" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

<optional>

<attribute name="table:protection-key">

<text/>

</attribute>

</optional>

</define>

#### Print

The table:print attribute specifies if a table is printed. It takes a Boolean value. If its value is true, the table is printed, if its value is false, the table is not printed. The default value is true. The table:print attribute will be overwritten by the table:display attribute described in section 15.8.14. That is, if the table is not displayed, it also will not be printed.

If the table is printed, the table range that actually is printed can be specified by table:print-ranges attribute (see section 8.1.1:Print Ranges). If this attribute is not existing, the used area of the table will be printed.

<define name="table-table-attlist" combine="interleave">

<optional>

<attribute name="table:print" a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Print Ranges

The table:print-ranges attribute specifies the print ranges of the table, i.e., the cells that should be printed. It contains a list of cell addresses or cell range addresses as described in section 8.3.1.

<define name="table-table-attlist" combine="interleave">

<optional>

<attribute name="table:print-ranges">

<ref name="cellRangeAddressList"/>

</attribute>

</optional>

</define>

#### Soft Page Breaks

The <text:soft-page-break> element represents a soft page break between two table rows. It may appear in front of <table:table-row> elements.

See section 2.3.1:Use Soft Page BreaksUse Soft Page Breaks for details regarding soft page breaks.

### Table Row

The <table:table-row> element represents a row in a table. It content are elements that specify the cells of the table row.

The <table:table-row> element is similar to the [XSL] <fo:table-row> element.

<define name="table-table-row">

<element name="table:table-row">

<ref name="table-table-row-attlist"/>

<oneOrMore>

<choice>

<ref name="table-table-cell"/>

<ref name="table-covered-table-cell"/>

</choice>

</oneOrMore>

</element>

</define>

#### Number of Rows Repeated

The table:number-rows-repeated attribute specifies the number of rows to which a row element applies. If two or more rows are adjoining, and have the same content and properties, and do not contain vertically merged cells, they may be described by a single <table:table-row> element that has a table:number-rows-repeated attribute with a value greater than 1.

<define name="table-table-row-attlist" combine="interleave">

<optional>

<attribute name="table:number-rows-repeated" a:defaultValue="1">

<ref name="positiveInteger"/>

</attribute>

</optional>

</define>

#### Row Style

A table row style stores the formatting properties of a table row, such as height and background color. A row style is defined by a <style:style> element with a family attribute value of table-row. The table row style can be either an automatic or a common style. It is referenced by the table row's table:style-name attribute.

<define name="table-table-row-attlist" combine="interleave">

<optional>

<attribute name="table:style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

</define>

#### Default Cell Style

The table:default-cell-style-name attribute specifies a default cell style. Cells contained in the row that don't have a table:style-style name attribute use this default cell style.

The attribute is applied to cells that are defined by a <table:table-cell> element. It is typically not applied to table cells that spreadsheet application may display in addition to those defined in the document.

<define name="table-table-row-attlist" combine="interleave">

<optional>

<attribute name="table:default-cell-style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

</define>

#### Visibility

The table:visibility attribute specifies whether the row is visible, filtered, or collapsed. Filtered and collapsed rows are not visible. Filtered rows are invisible, because a filter is applied to the table that does not select the table row. Collapsed rows have been made invisible by invisible in the UI directly.

<define name="table-table-row-attlist" combine="interleave">

<optional>

<attribute name="table:visibility" a:defaultValue="visible">

<ref name="table-visibility-value"/>

</attribute>

</optional>

</define>

<define name="table-visibility-value">

<choice>

<value>visible</value>

<value>collapse</value>

<value>filter</value>

</choice>

</define>

Example: Table with three rows and three columns

This example shows the OpenDocument code for a table with three rows and three columns. The first two rows of the table have a blue background.

<style:style style:name="Table 1" style:family="table">

<style:table-properties style:width="12cm"   
 fo:background-color="light-grey"/>

</style:style>

<style:style style:name="Col1" style:family="table-column">

<style:table-column-properties style:column-width="2cm"/>

</style:style>

<style:style style:name="Col2" style:family="table-column">

<style:table-column-properties style:column-width="4cm"/>

</style:style>

<style:style style:name="Col3" style:family="table-column">

<style:table-column-properties style:column-width="6cm"/>

</style:style>

<style:style style:name="Row1" style:family="table-row">

<style:table-row-properties fo:background-color="blue"/>

</style:style>

<table:table table:name="Table 1" table:style-name="Table 1">

<table:table-columns>

<table:table-column table:style-name="Col1"/>

<table:table-column table:style-name="Col2"/>

<table:table-column table:style-name="Col3"/>

</table:table-columns>

<table:table-rows>

<table:table-row table:style-name="Row1">

...

</table:table-row>

<table:table-row table:style-name="Row1">

...

</table:table-row>

<table:table-row>

...

</table:table-row>

<table:table-rows>

</table:table>

### Table Cell

The <table:table-cell> and <table:covered-table-cell> elements specify the content of a table cells. They are contained in table row elements. A table cell can contain paragraphs and other text content as well as sub tables. Table cells may be empty.

The <table:table-cell> element is very similar to the table cell elements of [XSL] and [HTML4], and the rules regarding cells that span several columns or rows that exist in HTML and XSL apply to the OpenDocument specification as well. This means that there are no <table:table-cell> elements in the row/column grid for positions that are covered by a merged cell, that is, that are covered by a cell that spans several columns or rows. The <table:covered-table-cell> element exists to be able to specify cells for such positions . It has to appear wherever a position in the row/column grid is covered by a cell that spans several rows or columns. Its position in the grid is calculated by a assuming a column and row span of 1 for all cells regardless whether they are specified by a <table:table-cell> or a <table:covered-table-cell> element. The <table:covered-table-cell> is especially used by spreadsheet applications, where it is a common use case that a covered cell contains content.

<define name="table-table-cell">

<element name="table:table-cell">

<ref name="table-table-cell-attlist"/>

<ref name="table-table-cell-attlist-extra"/>

<ref name="table-table-cell-content"/>

</element>

</define>

<define name="table-covered-table-cell">

<element name="table:covered-table-cell">

<ref name="table-table-cell-attlist"/>

<ref name="table-table-cell-content"/>

</element>

</define>

<define name="table-table-cell-content">

<optional>

<ref name="table-cell-range-source"/>

</optional>

<optional>

<ref name="office-annotation"/>

</optional>

<optional>

<ref name="table-detective"/>

</optional>

<zeroOrMore>

<ref name="text-content"/>

</zeroOrMore>

</define>

#### Number of Cells Repeated

The table:number-columns-repeated attribute specifies the number of successive columns in which a cell is repeated. It can be used to describe two or more adjoining cells with a single cell element, if they meet the following conditions:

* The cells contain the same content and properties.
* The cells are not merged horizontally or vertically.

In this case, a table:number-columns-repeated attribute must be used to specify the number of successive columns in which the cell is repeated. This attribute is specified with either the <table:table-cell> element or the <table:covered-table-cell> element.

<define name="table-table-cell-attlist" combine="interleave">

<optional>

<attribute name="table:number-columns-repeated" a:defaultValue="1">

<ref name="positiveInteger"/>

</attribute>

</optional>

</define>

#### Number of Rows and Columns Spanned

These attributes specify the number of rows and columns that a cell spans. These attributes can be used with the <table:table-cell> element only.

When a cell covers another cell because of a column or row span value greater than one, a <table:covered-table-cell> element must appear in the table to represent the covered cell.

<define name="table-table-cell-attlist-extra" combine="interleave">

<optional>

<attribute name="table:number-columns-spanned" a:defaultValue="1">

<ref name="positiveInteger"/>

</attribute>

</optional>

<optional>

<attribute name="table:number-rows-spanned" a:defaultValue="1">

<ref name="positiveInteger"/>

</attribute>

</optional>

</define>

#### Cell Style

A table cell style stores the formatting properties of a cell, such as the following:

* Background color
* Number format
* Vertical alignment
* Borders

The table cell style can be either an automatic or a common style. The style is specified with a table:style-name attribute. If a cell does not have a cell style assigned, the application checks if a the current row has a default cell style assigned. If the current row does not have a default cell assigned style as well, the application checks if the current column has a default cell style assigned.

<define name="table-table-cell-attlist" combine="interleave">

<optional>

<attribute name="table:style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

</define>

#### Cell Content Validation

The table:content-validation-name attribute specifies if a cell contains a validity check. The value of this attribute is the name of a <table:content-validation> element. If the attribute is not present, the cell may have arbitrary content.

<define name="table-table-cell-attlist" combine="interleave">

<optional>

<attribute name="table:content-validation-name">

<ref name="string"/>

</attribute>

</optional>

</define>

See section 8.5.3 for more information on cell content validation and the <table:cell-content-validation> element.

#### Formula

Formulas allow calculations to be performed within table cells. Every formula should begin with a namespace prefix specifying the syntax and semantics used within the formula. Typically, the formula itself begins with an equal (=) sign and can include the following components:

* Numbers.
* Text.
* Named ranges.
* Operators.
* Logical operators.
* Function calls.
* Addresses of cells that contain numbers. The addresses can be relative or absolute, see section 8.3.1. Addresses in formulas start with a “[“ and end with a “]”. See sections 8.3.1 and 8.3.1 for information about how to address a cell or cell range.

The following is an example of a simple formula:

=sum([.A1:.A5])

This formula calculates the sum of the values of all cells in the range “.A1:.A5”. The function is “sum”. The parameters are marked by a “(“ at the start and a “)” at the end. If a function contains more than one parameter, the parameters are separated by a “;”.

The following is a variation of the formula shown above:

=sum([.A1];[.A2];[.A3];[.A4];[.A5])

The result of this formula is the same. The components used in the formula depend on the application being used.

The table:formula attribute contains a formula for a table cell.

<define name="table-table-cell-attlist" combine="interleave">

<optional>

<attribute name="table:formula">

<ref name="string"/>

</attribute>

</optional>

</define>

In addition to this, the calculated value of the formula is available as well. One of the following attributes represents the current value of the cell:

* office:value
* office:date-value
* office:time-value
* office:boolean-value
* office:string-value

#### Matrix

When an application is performing spreadsheet calculations, a connected range of cells that contains values is called a matrix. If the cell range contains *m* rows and *n* columns, the matrix is called an *m x n* matrix. The smallest possible matrix is a *1 x 2* or *2 x 1* matrix with two adjacent cells. To use a matrix in a formula, include the cell range address of the matrix in the formula. In a matrix formula, only special matrix operations are possible.

The number of rows and columns that a matrix spans are represented by the table:number-matrix-rows-spanned and table:number-matrix-columns-spanned attributes, which are attached to the cell elements.

<define name="table-table-cell-attlist-extra" combine="interleave">

<optional>

<attribute name="table:number-matrix-columns-spanned">

<ref name="positiveInteger"/>

</attribute>

</optional>

<optional>

<attribute name="table:number-matrix-rows-spanned">

<ref name="positiveInteger"/>

</attribute>

</optional>

</define>

#### Value Type

The table:value-type attribute specifies the type of value that can appear in a cell. It may contain one of the following values:

* float, percentage or currency (numeric types)
* date
* time
* boolean
* string

<define name="table-table-cell-attlist" combine="interleave">

<optional>

<ref name="common-value-and-type-attlist"/>

</optional>

</define>

#### Cell Current Numeric Value

The office:value attribute specifies the current numeric value of a cell. This attribute is only evaluated for cells that contain the following data types:

* float
* percentage
* currency

#### Cell Current Currency

The tableoffice:currency attribute specifies the current currency value of a cell. The value of this attribute is usually currency information such as DEM or EUR. This attribute is only evaluated for cells whose data type is currency.

#### Cell Current Date Value

The office:date-value attribute specifies the current date value of a cell. This attribute is only evaluated for cells whose data type is date.

Some application support date and time values in addition to dates.

#### Cell Current Time Value

The office:time-value attribute specifies the current time value of a cell. This attribute is only evaluated for cells whose data type is time.

#### Cell Current Boolean Value

The office:boolean-value attribute specifies the current Boolean value of a cell. This attribute is only evaluated for cells whose data type is boolean.

#### Cell Current String Value

The office:string-value attribute specifies the current string value of a cell. This attribute is only evaluated for cells whose data type is string.

#### Table Cell Protection

The table:protected attribute protects the table cells. Users can not edit the content of a cell that is marked as protected.

<define name="table-table-cell-attlist" combine="interleave">

<optional>

<attribute name="table:protect" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

This attribute is not related to the table:protected attribute for table elements (see section 8.1.1) and the table:cell-protect attribute for table cell styles (see section 15.11.14).

## Advanced Table Model

### Column Description

Every column in a table has a column description element . It is similar to the [XSL] <fo:table-column> element, and its primary use is to reference a table column style that for instance specifies the table column's width.

<define name="table-table-column">

<element name="table:table-column">

<ref name="table-table-column-attlist"/>

<empty/>

</element>

</define>

#### Number of Columns Repeated

The attribute specifies the number of columns to which a column description applies. If two or more columns are adjoining, and have the same properties, this attribute allows to describe them with a single <table:table-column> element.

<define name="table-table-column-attlist" combine="interleave">

<optional>

<attribute name="table:number-columns-repeated" a:defaultValue="1">

<ref name="positiveInteger"/>

</attribute>

</optional>

</define>

#### Column Style

A table column style stores the formatting properties of a table column, such as width and background color. It is specified by a <style:style> element with a family attribute value of table-column and can be either an automatic or a common style. The style of a column is specified using a table:style-name attribute.

<define name="table-table-column-attlist" combine="interleave">

<optional>

<attribute name="table:style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

</define>

#### Visibility

The table:visibility attribute specifies whether the column is visible, filtered, or collapsed. See section 8.1.2 for more details.

<define name="table-table-column-attlist" combine="interleave">

<optional>

<attribute name="table:visibility" a:defaultValue="visible">

<ref name="table-visibility-value"/>

</attribute>

</optional>

</define>

#### Default Cell Style

The table:default-cell-style-name attribute specifies the default cell style. Cells that don't have a table:style-style name attribute use this style when there is no default cell style specified for the cell's row as well.

The attribute is applied to cells that are defined by a <table:table-cell> element. It is typically not applied to table cells that spreadsheet application may display in addition to those defined in the document.

<define name="table-table-column-attlist" combine="interleave">

<optional>

<attribute name="table:default-cell-style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

</define>

**Example: Table with three columns**

This example shows the OpenDocument code for a table with three columns.

<style:style style:name="Table 1" style:family="table">

<style:table-properties style:width="12cm"   
 fo:background-color="light-grey"/>

</style:style>

<style:style style:name="Col1" style:family="table-column">

<style:table-column-properties style:column-width="2cm"/>

</style:style>

<style:style style:name="Col2" style:family="table-column">

<style:table-column-properties style:column-width="4cm"/>

</style:style>

<style:style style:name="Col3" style:family="table-column">

<style:table-column-properties style:column-width="6cm"/>

</style:style>

<table:table table:name="Table 1" table:style-name="Table 1">

<table:table-columns>

<table:table-column table:style-name="Col1"/>

<table:table-column table:style-name="Col2"/>

<table:table-column table:style-name="Col3"/>

</table:table-columns>

...

</table:table>

### Header Columns

For accessibility purposes, header information is needed. Therefore, any columns designated as headers by the author must be tagged as such by encapsulating them within a <table:table-header-columns> element. Using style information only to designate header columns is insufficient.

If a table does not fit on a single page, table columns that are included in a <table:table-header-columns> element are automatically repeated on every page. A table must not contain more than one <table:table-header-columns> element, and a <table:table-columns> must not follow another <table:table-columns> element, with the only exception of tables that contain grouped columns (see 8.2.3). Such tables may contain more than one <table:table-header-columns> element, provided that they are contained in different column groups and the columns contained in the elements are adjacent.

Applications that do not support header columns have to process header column descriptions the same way as non header column descriptions.

The <table:table-header-columns> and <table:table-columns> element are very similar to [HTML4]'s <THEAD> and <TBODY> elements for rows.

<define name="table-table-header-columns">

<element name="table:table-header-columns">

<oneOrMore>

<ref name="table-table-column"/>

</oneOrMore>

</element>

</define>

<define name="table-table-columns">

<element name="table:table-columns">

<oneOrMore>

<ref name="table-table-column"/>

</oneOrMore>

</element>

</define>

### Column Groups

Adjacent table columns can be grouped with the element. Every group can contain a new group, columns, and column headers. A column group can be visible or hidden. Column groups can for instance used by spreadsheet applications to group columns that are summarized, so that the individual columns that contribute to the sum can be made invisible easily, but the sum remains visible.

If a set of header columns and a column group overlap, the header column group breaks the column header set. That is, the <table:table-column-group> may contain <table:table-header-columns> elements, but not vice versa.

<define name="table-table-column-group">

<element name="table:table-column-group">

<ref name="table-table-column-group-attlist"/>

<ref name="table-columns-and-groups"/>

</element>

</define>

#### Display

The table:display attribute specifies whether or not the group is visible.

<define name="table-table-column-group-attlist" combine="interleave">

<optional>

<attribute name="table:display" a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

### Header Rows

For accessibility purposes, header information is needed. Therefore, any rows designated as headers by the author must be tagged as such by encapsulating them within a <table:table-header-rows> element. Using style information only to designate header rows is insufficient.

If a table does not fit on a single page, table rows that are included in a <table:table-header-rows> element are automatically repeated on every page. A table must not contain more than one <table:table-header-rows> element. The one exception to this is a table that contains grouped rows (see 8.2.5). Such a table may contain more than one <table:table-header-rows> element, provided that they are contained in different row groups and the rows contained in the elements are adjacent.

Applications that do not support header rows have to process header rows the same way as non header rows.

The <table:table-header-rows> and <table:table-rows> element are very similar to [HTML4]'s <THEAD> and <TBODY> elements.

<define name="table-table-header-rows">

<element name="table:table-header-rows">

<oneOrMore>

<optional>

<ref name="text-soft-page-break"/>

</optional>

<ref name="table-table-row"/>

</oneOrMore>

</element>

</define>

<define name="table-table-rows">

<element name="table:table-rows">

<oneOrMore>

<optional>

<ref name="text-soft-page-break"/>

</optional>

<ref name="table-table-row"/>

</oneOrMore>

</element>

</define>

### Row Groups

Adjacent table rows can be grouped with the <table:table-row-group> element. Every group can contain a new group, rows, and row headers. A row group can be visible or hidden. Row groups can for instance used by spreadsheet applications to group rows that are summarized, so that the individual rows that contribute to the sum can be made invisible easily, but the sum remains visible.

If a set of header rows and a row group overlap, the header row group breaks the row header set. That is, the <table:table-row-group> may contain <table:table-header-rows> elements, but not vice versa.

<define name="table-table-row-group">

<element name="table:table-row-group">

<ref name="table-table-row-group-attlist"/>

<ref name="table-rows-and-groups"/>

</element>

</define>

#### Display

The table:display attribute specifies whether or not the group is visible.

<define name="table-table-row-group-attlist" combine="interleave">

<optional>

<attribute name="table:display" a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

### Subtables

If a table cell only contains a single table but no paragraphs or other content, this table can be specified as subtable. It then occupies the whole cell and no other content can appear in this cell.

The borders of a subtable merge with the borders of the cell that it resides in. A subtable does not contain any formatting properties. A subtable is essentially a container for some additional table rows that integrate seamlessly with the parent table.

A nested table is turned into a subtable with the attribute table:is-subtable that is attached to the table element. A nested table that is not a specified to be a subtable appears as a table within a table, that is, it has borders distinct from those of the parent cell and respects the padding of the parent cell.

<define name="table-table-attlist" combine="interleave">

<optional>

<attribute name="table:is-sub-table" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

Example of Representation of subtable

In the OpenDocument schema, this table can be represented in either of the ways detailed in Sample 1 and Sample 2.

| A1 | B1 | C1 |
| --- | --- | --- |
| A2 | |  |  | | --- | --- | | .B2.A1 | .B2.B1 | | .B2.A2 | | | |

*Sample 1*

Using cells that span several rows, the table is specified as follows:

<style:style style:name="Table 1" style:family="table">

<style:table-properties style:width="12cm"

fo:background-color="light-grey"/>

</style:style>

<style:style style:name="Col1" style:family="table-column">

<style:table-column-properties style:column-width="2cm"/>

</style:style>

<style:style style:name="Col2" style:family="table-column">

<style:table-column-properties style:column-width="4cm"/>

</style:style>

<style:style style:name="Col3" style:family="table-column">

<style:table-column-properties style:column-width="6cm"/>

</style:style>

<style:style style:name="Row1" style:family="table-row">

<style:table-row-properties fo:background-color="grey"/>

</style:style>

<style:style style:name="Cell1" style:family="table-cell">

<style:table-cell-properties fo:background-color="grey"/>

</style:style>

<table:table table:name="Table 1" table:style-name="Table 1">

<table:table-columns>

<table:table-column table:style-name="Col1"/>

<table:table-column table:style-name="Col2"/>

<table:table-column table:style-name="Col3"/>

</table:table-columns>

<table:table-header-rows>

<table:table-row table:style-name="Row1">

<table:table-cell>

<text:p text:style="Table Caption">

A1

</text:p>

</table:table-cell>

<table:table-cell>

<text:p text:style="Table Caption">

B1

</text:p>

</table:table-cell>

<table:table-cell>

<text:p text:style="Table Caption">

C1

</text:p>

</table:table-cell>

</table:table-row>

</table:table-header-rows>

<table:table-rows>

<table:table-row>

<table:table-cell table:number-rows-spanned="2"

table:style-name="Cell1">

<text:p text:style="Table Body">

A2

</text:p>

</table:table-cell>

<table:table-cell>

<text:p text:style="Table Body">

.B2.A1

</text:p>

</table:table-cell>

<table:table-cell>

<text:p text:style="Table Body">

.B2.B1

</text:p>

</table:table-cell>

</table:table-row>

<table:table-row>

<table:covered-table-cell/>

<table:table-cell table:number-columns-spanned="2">

<text:p text:style="Table Body">

.B2.A2

</text:p>

</table:table-cell>

<table:covered-table-cell/>

</table:table-row>

</table:table-rows>

</table:table>

*Sample 2*

Using sub tables, the table is specified as follows:

<style:style style:name="Table 1" style:family="table">

<style:table-properties fo:width="12cm" fo:background-color="light-grey"/>

</style:style>

<style:style style:name="Col1" style:family="table-column">

<style:table-column-properties style:column-width="2cm"/>

</style:style>

<style:style style:name="Col2" style:family="table-column">

<style:table-column-properties style:column-width="4cm"/>

</style:style>

<style:style style:name="Col3" style:family="table-column">

<style:table-column-properties style:column-width="6cm"/>

</style:style>

<style:style style:name="Row1" style:family="table-row">

<style:table-row-properties fo:background-color="grey"/>

</style:style>

<style:style style:name="Cell1" style:family="table-cell">

<style:table-cell-properties fo:background-color="grey"/>

</style:style>

<table:table table:name="Table 1" table:style-name="Table 1">

<table:table-columns>

<table:table-column table:style-name="Col1"/>

<table:table-column table:style-name="Col2"/>

<table:table-column table:style-name="Col3"/>

</table:table-columns>

<table:table-header-rows>

<table:table-row table:style-name="Row1">

<table:table-cell>

<text:p text:style="Table Caption">

A1

</text:p>

</table:table.cell>

<table:table-cell>

<text:p text:style="Table Caption">

B1

</text:p>

</table:table-cell>

<table:table-cell>

<text:p text:style="Table Caption">

C1

</text:p>

</table:table-cell>

</table:table-row>

</table:table-header-rows>

<table:table-rows>

<table:table-row>

<table:table-cell table:style-name="Cell1">

<text:p text:style="Table Body">

A2

</text:p>

</table:table-cell>

<table:table-cell table:number-columns-spanned="2">

<table:table is-subtable="true">

<table:table-columns>

<table:table-column table:style-name="Col2"/>

<table:table-column table:style-name="Col3"/>

</table:table-columns>

<table:rows>

<table:row>

<table:table-cell>

<text:p text:style="Table Body">

.B2.A1

</text:p>

</table:table-cell>

<table:table-cell>

<text:p text:style="Table Body">

.B2.B1

</text:p>

</table:table-cell>

</table:table-row>

<table:table-row>

<table:table-cell

table:number-columns-spanned="2">

<text:p text:style="Table Body">

.B2.A2

</text:p>

</table:table-cell>

<table:covered-table-cell/>

</table:table-row>

</table:table-rows>

</table:table>

</table:table-cell>

<table:covered-table-cell/>

</table:table-row>

</table:table-rows>

</table:table>

## Advanced Tables

### Referencing Table Cells

To reference table cells so called cell addresses are used. The structure of a cell address is as follows:

1. The name of the table.
2. A dot (.).
3. An alphabetic value representing the column. The letter A represents column 1, B represents column 2, and so on. AA represents column 27, AB represents column 28, and so on.
4. A numeric value representing the row. The number 1 represents the first row, the number 2 represents the second row, and so on.

This means that A1 represents the cell in column 1 and row 1. B1 represents the cell in column 2 and row 1. A2 represents the cell in column 1 and row 2.

For example, in a table with the name SampleTable the cell in column 34 and row 16 is referenced by the cell address SampleTable.AH16. In some cases it is not necessary to provide the name of the table. However, the dot must be present. When the table name is not required, the address in the previous example is .AH16.

The structure of the address of a cell in a subtable is as follows:

1. The address of the cell that contains the subtable.
2. A dot (.).
3. The address of the cell in the subtable.

For example, to reference the cell in column 1 and row 1 in a subtable that is called Subtable, and that is in column 34 and row 16 of the table SampleTable, the address is SampleTable.AH16.A1.

If the name of the table contains blanks, dots (.) or apostrophes ('), the name must be quoted with apostrophes ('). Any apostrophes in the name must be escaped by doubling ('').

E.g. 'Tom''s Table'.A1 for the cell A1 in the table named Tom's Table.

#### Absolute and relative cell addressing

Cells can be referenced by using either absolute addresses or relative addresses. When an operation is performed on a table cell, for example when a formula is copied, absolute cell references do not change; In contrast to this, relative cell references are adapted to the address of target cell of the copy operation. The previous example uses relative addressing.

To create an absolute address, a dollar sign ($) has to be placed before each table name, column reference, and row reference. For example, the absolute address of the previous example is $SampleTable.$AH$16. Absolute and relative references can be mixed within a single cell address. For example, SampleTable.AH$16 refers to a relative table and column, but to an absolute row. Absolute addresses must contain a table name. The differentiation between absolute and relative addressing is only necessary in some situations. Where a differentiation is not required, a cell reference without the dollar signs is used.

<define name="cellAddress">

<data type="string">

<param name="pattern">($?([^\. ']+|'([^']|'')+'))?\.$?[A-Z]+$?[0-9]+</param>

</data>

</define>

#### Cell Range Address

A cell range is a number of adjacent cells forming a rectangular shape. The rectangle stretches from the cell on the top left to the cell on the bottom right.

A cell range address references a cell range. It is constructed as follow:

1. The address of the cell at the top left of the range.
2. A colon (:).
3. The address of the cell at the bottom right of the range.

For example, the address .A1:.B2 references the cell range of cells from column 1 and row 1 to column 2 and row 2. The smallest range one can specify is a single cell. In this case, the range address is the same as the cell address.

<define name="cellRangeAddress">

<data type="string">

<param name="pattern">($?([^\. ']+|'([^']|'')+'))?\.$?[A-Z]+$?[0-9]+(:($?([^\. ']+|'([^']|'')+'))?\.$?[A-Z]+$?[0-9]+)?</param>

</data>

</define>

#### Cell Range Address List

A cell range address list is a list of cell ranges and cell addresses. Each item in the list is separated by a space. If table names used in the list contain a blank character, the table name has to be quoted within apostrophes (').

<define name="cellRangeAddressList">

<!-- Value is a space separated list of "cellRangeAddress" patterns -->

<data type="string"/>

</define>

### Linked Tables

If a table is linked to an original table, the information about the source table is contained in a <table:table-source> element. The attributes that may be associated with the <table:table-source> element are:

* Mode
* Table name
* URL
* Filter name
* Filter options
* Refresh delay

<define name="table-table-source">

<element name="table:table-source">

<ref name="table-table-source-attlist"/>

<ref name="table-linked-source-attlist"/>

<empty/>

</element>

</define>

#### Mode

The table:mode attribute specifies what data should be copied from the source table to the destination table. If the attribute's value is “copy-all” formulas and styles are copied. If the attribute's value is “copy-results-only”, only formula results and non calculated cell content will be copied.

<define name="table-table-source-attlist" combine="interleave">

<optional>

<attribute name="table:mode" a:defaultValue="copy-all">

<choice>

<value>copy-all</value>

<value>copy-results-only</value>

</choice>

</attribute>

</optional>

</define>

#### Table Name

The table:table-name attribute specifies the name of the table in the original document. If the table name is not specified, the first table in the document is used.

<define name="table-table-source-attlist" combine="interleave">

<optional>

<attribute name="table:table-name">

<ref name="string"/>

</attribute>

</optional>

</define>

#### URL

The original table is specified by a an XLink, where the xlink:href attribute specifies the URL of the document containing the original table.

<define name="table-linked-source-attlist" combine="interleave">

<optional>

<attribute name="xlink:type" a:defaultValue="simple">

<value>simple</value>

</attribute>

</optional>

<optional>

<attribute name="xlink:actuate" a:defaultValue="onRequest">

<value>onRequest</value>

</attribute>

</optional>

<attribute name="xlink:href">

<ref name="anyURI"/>

</attribute>

</define>

#### Filter Name

The table:filter-name attribute specifies the file type of the document containing the original table. The value of this attribute is application-specific.

<define name="table-linked-source-attlist" combine="interleave">

<optional>

<attribute name="table:filter-name">

<ref name="string"/>

</attribute>

</optional>

</define>

#### Filter Options

The table:filter-options attribute specifies optional settings about the file type. The value of this attribute is application-specific.

<define name="table-linked-source-attlist" combine="interleave">

<optional>

<attribute name="table:filter-options">

<ref name="string"/>

</attribute>

</optional>

</define>

#### Refresh Delay

The table:refresh-delay attribute specifies the time delay between refresh actions for the linked table.

<define name="table-linked-source-attlist" combine="interleave">

<optional>

<attribute name="table:refresh-delay">

<ref name="duration"/>

</attribute>

</optional>

</define>

### Scenario Tables

A scenario is an area of a table where data from other, so called scenario tables, is linked to temporarily. If several scenarios are defined for the same area, an user might choose between the scenarios. Whether a scenario table is visible itself is controlled by table's style. Only one scenario table can be active per table.

A table that contains a <table:scenario> represents a scenario table. The name of the table and the name of the scenario are the same. The scenario is displayed in the regular table preceding the scenario table. If a scenario table is existing for a table, a scenario is displayed on that table automatically. These means the the existence of a scenario table implies the existence of a scenario.

The attributes that may be associated with this element are:

* Scenario Ranges
* Is Active
* Display Border
* Border Color
* Copy Back
* Copy Styles
* Copy Formulas
* Comment
* Protected

<define name="table-scenario">

<element name="table:scenario">

<ref name="table-scenario-attlist"/>

<empty/>

</element>

</define>

#### Scenario Ranges

The table:scenario-ranges attribute specifies the table range that is displayed as a scenario. The value of this attribute is a list of cell range addresses.

<define name="table-scenario-attlist" combine="interleave">

<attribute name="table:scenario-ranges">

<ref name="cellRangeAddressList"/>

</attribute>

</define>

#### Is Active

The table:is-active attribute specifies whether or not the scenario that belongs to the scenario table is active.

<define name="table-scenario-attlist" combine="interleave">

<attribute name="table:is-active">

<ref name="boolean"/>

</attribute>

</define>

#### Display Border

The table:display-border attribute specifies whether or not to display a border around the scenario that belongs to the scenario table.

<define name="table-scenario-attlist" combine="interleave">

<optional>

<attribute name="table:display-border" a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Border Color

The table:border-color attribute specifies the color of the border that is displayed around the scenario that belongs to the scenario table.

<define name="table-scenario-attlist" combine="interleave">

<optional>

<attribute name="table:border-color">

<ref name="color"/>

</attribute>

</optional>

</define>

#### Copy Back

The table:copy-back attribute specifies whether or not data is copied back into the scenario table if another scenario is activated.

<define name="table-scenario-attlist" combine="interleave">

<optional>

<attribute name="table:copy-back" a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Copy Styles

The table:copy-styles attribute specifies whether or not styles are copied from the scenario table to the destination table together with the data.

<define name="table-scenario-attlist" combine="interleave">

<optional>

<attribute name="table:copy-styles" a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Copy Formulas

The attribute specifies whether or not formulas are copied from the scenario table to the destination table. The value of this attribute can be true or false. If the value is true, the formulas are copied. If the value is false, only the values resulting from the formulas are copied.

<define name="table-scenario-attlist" combine="interleave">

<optional>

<attribute name="table:copy-formulas" a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Comment

The table:comment attribute contains a comment about the scenario.

<define name="table-scenario-attlist" combine="interleave">

<optional>

<attribute name="table:comment">

<ref name="string"/>

</attribute>

</optional>

</define>

#### Protected

The table:protected attribute specifies whether or not the data that is displayed within the scenario is protected from being edited. The attribute is only evaluated if the table on which the scenario displayed is also protected (see section 8.1.1).

<define name="table-scenario-attlist" combine="interleave">

<optional>

<attribute name="table:protected">

<ref name="boolean"/>

</attribute>

</optional>

</define>

### Shapes

The <table:shapes> element contains all graphic shapes with an anchor on the table this element is a child of. It is a container element and does not have any associated attributes.

<define name="table-shapes">

<element name="table:shapes">

<oneOrMore>

<ref name="shape"/>

</oneOrMore>

</element>

</define>

## Advanced Table Cells

### Linked Table Cells

A cell range can be linked to a database range or named range of another file. In this case the information about the original database range or named range is contained in a <table:cell-range-source> element that is contained in the element of the first cell of the range. The attributes that may be associated with this element are:

* Name
* Last size
* URL
* Filter name
* Filter options
* Refresh delay

<define name="table-cell-range-source">

<element name="table:cell-range-source">

<ref name="table-table-cell-range-source-attlist"/>

<ref name="table-linked-source-attlist"/>

<empty/>

</element>

</define>

#### Name

The table:name attribute specifies the name of the source database range or named range.

<define name="table-table-cell-range-source-attlist" combine="interleave">

<attribute name="table:name">

<ref name="string"/>

</attribute>

</define>

#### Last Size

The table:last-column-spanned and table:last-row-spanned attributes specify the last known size of the range. If the size of the range is changed since the last operation, the values of these attributes are incorrect.

<define name="table-table-cell-range-source-attlist" combine="interleave">

<attribute name="table:last-column-spanned">

<ref name="positiveInteger"/>

</attribute>

<attribute name="table:last-row-spanned">

<ref name="positiveInteger"/>

</attribute>

</define>

#### URL, Filter Name, Filter Options and Refresh Delay

The attributes xlink:href, xlink:type, xlink:actuate, table:filter-name and table:filter-options are the same as for linked tables. See section 8.3.2 for details.

### Cell Annotation

The OpenDocument format allows annotation to appear within table cells. See section 12.1 for details on annotations.

### Detective

The <table:detective> element has two purposes. One the one hand, it contains information about relations that exist between table cells because of formulas and that should be highlighted in the UI. On the other hand, the element contains information about cells that are highlighted currently in the UI either because of the relations mentioned above or because of error conditions.

<define name="table-detective">

<element name="table:detective">

<zeroOrMore>

<ref name="table-highlighted-range"/>

</zeroOrMore>

<zeroOrMore>

<ref name="table-operation"/>

</zeroOrMore>

</element>

</define>

The elements that can be contained in the <table:detective> element are:

* Detective Operation
* Highlighted range

### Detective Operation

The <table:operation> element specifies that certain relations that exist between the cell the element is a child of and other cells should be made visible or invisible in the UI. One and the same detective operation can be applied multiple times to the same cell. In this case, the second operation is applied to the resulting cells of the first operation and so on. This means that an operation not necessarily is applied to the cell the operation is defined in, but also to other cells, and that it therefor can interact with operations defined in other cells. This especially applies to operations that make relations invisible. To get a determinate behavior, operations have an index and are applied in the order of that index. The attributes associated with the <table:operation> element are:

* Name
* Index

<define name="table-operation">

<element name="table:operation">

<ref name="table-operation-attlist"/>

<empty/>

</element>

</define>

#### Name

The table:name attribute specifies the name of the detective operation. Possible names are trace-dependents , remove-dependents, trace-precedents, remove-precedents and trace-errors. trace-dependents and remove-dependents displays or hides cells that use the value of the current cell in their formula. Trace-precedents and remove-precedents displays or hides cells whose value is used in the formula of the current cell. Trace-errors displays cells that cause an error while calculating the result of the current cell's formula.

<define name="table-operation-attlist" combine="interleave">

<attribute name="table:name">

<choice>

<value>trace-dependents</value>

<value>remove-dependents</value>

<value>trace-precedents</value>

<value>remove-precedents</value>

<value>trace-errors</value>

</choice>

</attribute>

</define>

#### Index

The table:index attribute specifies the the order in which detective operations are applied.

<define name="table-operation-attlist" combine="interleave">

<attribute name="table:index">

<ref name="nonNegativeInteger"/>

</attribute>

</define>

### Highlighted Range

The <table:highlighted-range> element specifies a cell range that is highlighted in the UI either because of detective operations described above or because it contains an error or invalid data.

The information contained in this element is not guaranteed to be up to date but reflects the state that at the time the detective operations or error conditions have been calculated.

The attributes that can be associated with the <table:highlighted-range> element are:

* Cell Range Address
* Direction
* Contains Error
* Marked Invalid

<define name="table-highlighted-range">

<element name="table:highlighted-range">

<choice>

<group>

<ref name="table-highlighted-range-attlist"/>

</group>

<group>

<ref name="table-highlighted-range-attlist-invalid"/>

</group>

</choice>

<empty/>

</element>

</define>

#### Cell Range Address

The table:cell-range-address attribute contains the address of a range that is highlighted currently.

<define name="table-highlighted-range-attlist" combine="interleave">

<optional>

<attribute name="table:cell-range-address">

<ref name="cellRangeAddress"/>

</attribute>

</optional>

</define>

#### Direction

The table:direction attribute specifies the direction of the relation between this cell and the highlighted range. The direction for instance might be visualized by an arrow.

<define name="table-highlighted-range-attlist" combine="interleave">

<attribute name="table:direction">

<choice>

<value>from-another-table</value>

<value>to-another-table</value>

<value>from-same-table</value>

</choice>

</attribute>

</define>

#### Contains Error

The table:contains-error attribute specifies whether or not the cell range contains an error.

<define name="table-highlighted-range-attlist" combine="interleave">

<optional>

<attribute name="table:contains-error" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Marked Invalid

The table:marked-invalid attribute specifies whether or not the current cell is marked invalid. This attribute cannot be used together with any other attributes.

<define name="table-highlighted-range-attlist-invalid" combine="interleave">

<attribute name="table:marked-invalid">

<ref name="boolean"/>

</attribute>

</define>

## Spreadsheet Document Content

### Document Protection

The structure of a spreadsheet document may be protected by using the table:structure-protected attribute, so that users can not insert, delete, move or rename the tables in the document. The optional table:protection-key attribute may be used to specify a password that prevents users from resetting the table protection flag to allow editing. To avoid saving the password directly into the XML file, only a hash value of the password is stored.

<define name="office-spreadsheet-attlist" combine="interleave">

<optional>

<attribute name="table:structure-protected" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

<optional>

<attribute name="table:protection-key">

<ref name="string"/>

</attribute>

</optional>

</define>

### Calculation Settings

Spreadsheet documents contain settings that affect the calculation of formulas, for example the null date or iteration settings. These settings must be saved in the document in the <table:calculation-settings> element.

<define name="table-calculation-settings">

<element name="table:calculation-settings">

<ref name="table-calculation-setting-attlist"/>

<optional>

<ref name="table-null-date"/>

</optional>

<optional>

<ref name="table-iteration"/>

</optional>

</element>

</define>

The attributes that may be associated with the <table:calculation-settings> element are:

* Case sensitive
* Precision as shown
* Search criteria must apply to whole cell
* Automatic find labels
* Use regular expression
* Null year
* Null date
* Iteration

#### Case Sensitive

The table:case-sensitive attribute specifies whether or not to distinguish between upper and lower case when comparison operators are applied to cell content.

<define name="table-calculation-setting-attlist" combine="interleave">

<optional>

<attribute name="table:case-sensitive" a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Precision as Shown

The table:precision-as-shown attribute specifies whether to perform a calculation using the rounded values displayed in the spreadsheet or using all of the digits in a number. If the value of this attribute is true, calculation are performed using the rounded values displayed in the spreadsheet. If the value of this attribute is false, calculations are performed using all of the digits in the number, but the result is still displayed as a rounded number.

<define name="table-calculation-setting-attlist" combine="interleave">

<optional>

<attribute name="table:precision-as-shown" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Search Criteria Must Apply to Whole Cell

The table:search-criteria-must-apply-to-whole-cell attribute specifies whether or not the specified search criteria, according to the regular expression used, must apply to the entire cell contents.

<define name="table-calculation-setting-attlist" combine="interleave">

<optional>

<attribute name="table:search-criteria-must-apply-to-whole-cell"

a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Automatic Find Labels

The table:automatic-find-labels attribute specifies whether or not to automatically find the labels of rows and columns.

<define name="table-calculation-setting-attlist" combine="interleave">

<optional>

<attribute name="table:automatic-find-labels" a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Use Regular Expressions

The table:use-regular-expressions attribute specifies whether regular expressions are enabled for character string comparisons and when searching.

<define name="table-calculation-setting-attlist" combine="interleave">

<optional>

<attribute name="table:use-regular-expressions"

a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Null Year

The table:null-year attribute specifies the start year for year values that contain only two digits. All two digit year values are interpreted as a year that equals or follows the start year.

<define name="table-calculation-setting-attlist" combine="interleave">

<optional>

<attribute name="table:null-year" a:defaultValue="1930">

<ref name="positiveInteger"/>

</attribute>

</optional>

</define>

#### Null Date

The <table:null-date> element specifies the null date. The null date is the date that results in the value “0” if a date value is converted into a numeric value. The null date is specified in the element's table:date-value attribute. Commonly used values are 12/30/1899, 01/01/1900, and 01/01/1904

<define name="table-null-date">

<element name="table:null-date">

<optional>

<attribute name="table:value-type" a:defaultValue="date">

<ref name="valueType"/>

</attribute>

</optional>

<optional>

<attribute name="table:date-value"

a:defaultValue="1899-12-30">

<ref name="date"/>

</attribute>

</optional>

<empty/>

</element>

</define>

#### Iteration

The <table:iteration> element enables formulas with iterative (or cyclic) references to be calculated after a specific number of iterations. Formulas with iterative references are repeated until the problem is solved. If this iterative calculations are not enabled, a formula with an iterative reference in a table causes an error message.

Iterative calculations are enabled and disabled with the table:status attribute. If iterative calculations are enabled, the table:steps attribute specifies the maximum number of iterations allowed. The table:maximum-difference attribute specifies the maximum difference allowed between two calculation results. The iteration is stopped if the result is less than the value of this attribute.

<define name="table-iteration">

<element name="table:iteration">

<optional>

<attribute name="table:status" a:defaultValue="disable">

<choice>

<value>enable</value>

<value>disable</value>

</choice>

</attribute>

</optional>

<optional>

<attribute name="table:steps" a:defaultValue="100">

<ref name="positiveInteger"/>

</attribute>

</optional>

<optional>

<attribute name="table:maximum-difference"

a:defaultValue="0.001">

<ref name="double"/>

</attribute>

</optional>

<empty/>

</element>

</define>

### Table Cell Content Validations

Table cell content validations specify validation rules for the content of table cells. The <table:content-validation> element specifies such a validation rule. All validation rules that exist in a document are contained <table:content-validations> element. The validation rules themselves are named and referenced from the table cell by its name.

<define name="table-content-validations">

<element name="table:content-validations">

<oneOrMore>

<ref name="table-content-validation"/>

</oneOrMore>

</element>

</define>

<define name="table-content-validation">

<element name="table:content-validation">

<ref name="table-validation-attlist"/>

<optional>

<ref name="table-help-message"/>

</optional>

<optional>

<choice>

<ref name="table-error-message"/>

<group>

<ref name="table-error-macro"/>

<optional>

<ref name="office-event-listeners"/>

</optional>

</group>

</choice>

</optional>

</element>

</define>

The attributes that may be associated with the <table:content-validation> element are:

* Name
* Condition
* Base cell address
* Allow empty cell
* Display list

#### Name

The table:name attribute specifies the name of the content validation. It is used to reference the validation rule from the cell the rule should applied to. The name is created automatically by the application.

<define name="table-validation-attlist" combine="interleave">

<attribute name="table:name">

<ref name="string"/>

</attribute>

</define>

#### Condition

The table:condition attribute specifies the condition that must evaluate to “true” for all cells the validation rule is applied to. The value of this attribute should be a namespace prefix, followed by a Boolean expression.

A typical syntax of the expression may be similar to the XPath syntax. The following are valid conditions:

* Condition ::= ExtendedTrueCondition | TrueFunction 'and' TrueCondition
* TrueFunction ::= cell-content-is-whole-number() | cell-content-is-decimal-number() | cell-content-is-date() | cell-content-is-time() | cell-content-is-text()
* ExtendedTrueCondition ::= ExtendedGetFunction | cell-content-text-length() Operator Value
* TrueCondition ::= GetFunction | cell-content() Operator Value
* GetFunction ::= cell-content-is-between(Value, Value) | cell-content-is-not-between(Value, Value)
* ExtendedGetFunction ::= cell-content-text-length-is-between(Value, Value) | cell-content-text-length-is-not-between(Value, Value) | cell-content-is-in-list( StringList )
* Operator ::= '<' | '>' | '<=' | '>=' | '=' | '!='
* Value ::= NumberValue | String | Formula
* StringList ::= StringList ';' String | String
* A Formula is a formula without an equals (=) sign at the beginning. See section 8.1.3 for more information.
* A String comprises one or more characters surrounded by quotation marks.
* A NumberValue is a whole or decimal number. It must not contain comma separators for numbers of 1000 or greater.

<define name="table-validation-attlist" combine="interleave">

<optional>

<attribute name="table:condition">

<ref name="string"/>

</attribute>

</optional>

</define>

#### Base Cell Address

The table:base-cell-address attribute specifies the address of the base cell for relative addresses in formulas that occur within a condition. This attribute is only necessary when the condition contains a formula. The value of this attribute must be an absolute cell address that contains a table name.

<define name="table-validation-attlist" combine="interleave">

<optional>

<attribute name="table:base-cell-address">

<ref name="cellAddress"/>

</attribute>

</optional>

</define>

#### Allow Empty Cell

The table:allow-empty-cell attribute specifies whether or not a cell can be empty.

<define name="table-validation-attlist" combine="interleave">

<optional>

<attribute name="table:allow-empty-cell" a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Display List

The table:display-list attribute specifies whether a list of values that occurs within a condition is displayed in the UI wile entering a cell value. The value of this attribute can be none, unsorted or sort-ascending.

* none: the list values are not displayed.
* unsorted: the list values are displayed in the order they occur in the condition.
* sort-ascending: the list values are displayed in ascending order.

<define name="table-validation-attlist" combine="interleave">

<optional>

<attribute name="table:display-list" a:defaultValue="unsorted">

<choice>

<value>none</value>

<value>unsorted</value>

<value>sort-ascending</value>

</choice>

</attribute>

</optional>

</define>

#### Help Message

The <table:help-message> element specifies a message to display if a user selects the cell. The element has an optional table:title attribute that specifies a title of the help message. It further has an optional table:display attribute that can be used to suppress the display of the message.

<define name="table-help-message">

<element name="table:help-message">

<optional>

<attribute name="table:title">

<ref name="string"/>

</attribute>

</optional>

<optional>

<attribute name="table:display" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

<zeroOrMore>

<ref name="text-p"/>

</zeroOrMore>

</element>

</define>

#### Error Message

The <table:error-message> element specifies a message to display if a user tries to enter invalid content into a cell i.e., content where the validation rule's condition evaluates to “false”. The element has an optional table:title attribute that specifies a title of the help message. It further has an optional table:display attribute that can be used to suppress the display of the message. The table:message-type attribute, that can take the values stop, warning, or information, specifies whether the message should be displayed as error (stop), warning (warning) or information only (information). In case the message is displayed as error, the operation that caused the validation check (for instance a cursor travel to leave the cell) is stopped.

<define name="table-error-message">

<element name="table:error-message">

<optional>

<attribute name="table:title">

<ref name="string"/>

</attribute>

</optional>

<optional>

<attribute name="table:display" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

<optional>

<attribute name="table:message-type" a:defaultValue="stop">

<choice>

<value>stop</value>

<value>warning</value>

<value>information</value>

</choice>

</attribute>

</optional>

<zeroOrMore>

<ref name="text-p"/>

</zeroOrMore>

</element>

</define>

#### Error Macro

As an alternative to displaying a message, a macro might be called if a cell contains invalid content. The macro in this case is specified by an <office:event-listeners> element as specified in section 12.4. The event name must be one that specifies an event that is called on invalid user input.

In addition to the <office:event-listeners> element, the <table:error-macro> element specifies whether the macro should be executed or not.

<define name="table-error-macro">

<element name="table:error-macro">

<optional>

<attribute name="table:execute" a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</element>

</define>

### Label Ranges

Label ranges can be used to assign names to rows and columns, or to parts of rows and columns, where the names themselves are specified as the content of table cells. More precisely, the label range element <table:label-range> specifies a label cell range which contain the labels, and data cell range which specifies the rows or columns whose content is referenced by the labels.

There are two types of label ranges.

* Label ranges for columns
* Label ranges for rows.

The data cell range should have the same height and vertical position like the label cell range if row labels are specified, or should have the same width and horizontal position like the label range if column labels are specified. For information on defining a cell range, see section 8.3.1.

Labels can be used within formula like any other name. All label ranges that exist in a document are contained within a single <table:label-ranges> element.

<define name="table-label-ranges">

<element name="table:label-ranges">

<zeroOrMore>

<ref name="table-label-range"/>

</zeroOrMore>

</element>

</define>

<define name="table-label-range">

<element name="table:label-range">

<ref name="table-label-range-attlist"/>

<empty/>

</element>

</define>

#### Label Cell Range Address

The table:label-cell-range-address attribute specifies the cell range address of the labels.

<define name="table-label-range-attlist" combine="interleave">

<attribute name="table:label-cell-range-address">

<ref name="cellRangeAddress"/>

</attribute>

</define>

#### Data Cell Range Address

The table:data-cell-range-address attribute specifies the cell range address of the data.

<define name="table-label-range-attlist" combine="interleave">

<attribute name="table:data-cell-range-address">

<ref name="cellRangeAddress"/>

</attribute>

</define>

#### Orientation

The table:orientation attribute specifies the orientation of the label range. This attribute can have a value of column or row.

<define name="table-label-range-attlist" combine="interleave">

<attribute name="table:orientation">

<choice>

<value>column</value>

<value>row</value>

</choice>

</attribute>

</define>

### Named Expressions

The named expressions element <table:named-expressions> contains a collection of assignments of names to expressions, so that the names can be use to refer to the expression.

The following expression can get names:

* cell ranges.
* Other expressions, for example, parts of a formula.

<define name="table-named-expressions">

<element name="table:named-expressions">

<zeroOrMore>

<choice>

<ref name="table-named-range"/>

<ref name="table-named-expression"/>

</choice>

</zeroOrMore>

</element>

</define>

#### Named Range

The named range element <table:named-range> specifies a cell range that has a name assigned. For information on defining a cell range, see section 8.3.1.

The table:name attribute specifies the name of the range, and the table:cell-range-address attribute its address. The address can be either absolute or relative. If the cell range address is relative, the table:base-cell-address attribute must exist additionally. It specifies the base cell address for the cell range. This address must be absolute. Therefore a table name in the address is required, but the dollar signs that indicate an absolute address can be omitted.

An additional table:range-usable-as attribute specifies whether the name of the range can be used within the specification of a print range, a filter, a repeating row, or a repeat column. The value of this attribute can be either:

* none, or
* a space-separated list that consists of any of the values print-range, filter, repeat-row or repeat-column.

<define name="table-named-range">

<element name="table:named-range">

<ref name="table-named-range-attlist"/>

<empty/>

</element>

</define>

<define name="table-named-range-attlist" combine="interleave">

<attribute name="table:name">

<ref name="string"/>

</attribute>

<attribute name="table:cell-range-address">

<ref name="cellRangeAddress"/>

</attribute>

<optional>

<attribute name="table:base-cell-address">

<ref name="cellAddress"/>

</attribute>

</optional>

<optional>

<attribute name="table:range-usable-as" a:defaultValue="none">

<choice>

<value>none</value>

<list>

<oneOrMore>

<choice>

<value>print-range</value>

<value>filter</value>

<value>repeat-row</value>

<value>repeat-column</value>

</choice>

</oneOrMore>

</list>

</choice>

</attribute>

</optional>

</define>

#### Named Expression

The named expression element <table:named-expression> contains an expression with a name, for example, a part of a formula.

The table:name attribute specifies the name of the expression, and the table:expression attribute the expression itself. The expressions do not support the equal (=) sign as the first character. If the expression contains a named range or another named expression, the named range or named expression must be specified first, before the containing expression. If the expression contains a relative cell range address, the table:base-cell-address attribute must exist additionally. It specifies the base cell address for the cell range. This address must be absolute. Therefore a table name in the address is required, but the dollar signs that indicate an absolute address can be omitted.

<define name="table-named-expression">

<element name="table:named-expression">

<ref name="table-named-expression-attlist"/>

<empty/>

</element>

</define>

<define name="table-named-expression-attlist" combine="interleave">

<attribute name="table:name">

<ref name="string"/>

</attribute>

<attribute name="table:expression">

<ref name="string"/>

</attribute>

<optional>

<attribute name="table:base-cell-address">

<ref name="cellAddress"/>

</attribute>

</optional>

</define>

Example: Named expressions element with a named range and a named expression

<table:named-expressions>

<table:named-range table:name="sample1" table:cell-range-address=".C4"

table:base-cell-address="sampletable.F1" table:area-type="none"/>

<table:named-range table:name="sample2"

table:cell-range-address=".$D$3:.$K$8"

table:area-type="print-range filter"/>

<table:named-expression table:name="sample3"

table:expression="sum([.A1:.B3])"/>

</table:named-expressions>

## Database Ranges

A database range is a named area in a table where database operations, but also some other kind of operations like filtering and sorting, can be performed. The Database Ranges element <table:database-ranges> contains a collection of all database ranges defined in a document.

<define name="table-database-ranges">

<element name="table:database-ranges">

<zeroOrMore>

<ref name="table-database-range"/>

</zeroOrMore>

</element>

</define>

### Database Range

The <table:database-range> defines a single database range.

<define name="table-database-range">

<element name="table:database-range">

<ref name="table-database-range-attlist"/>

<optional>

<choice>

<ref name="table-database-source-sql"/>

<ref name="table-database-source-table"/>

<ref name="table-database-source-query"/>

</choice>

</optional>

<optional>

<ref name="table-filter"/>

</optional>

<optional>

<ref name="table-sort"/>

</optional>

<optional>

<ref name="table-subtotal-rules"/>

</optional>

</element>

</define>

#### Database Range Name

The table:name attribute specifies the name of the database range on which to perform operations. Within a single document, only one database range is allowed to have no name. This database range is usually automatically created by the application and is used to filter or sort data in a cell ranges without the user explicitly creating a database range.

<define name="table-database-range-attlist" combine="interleave">

<optional>

<attribute name="table:name">

<ref name="string"/>

</attribute>

</optional>

</define>

#### Is Selection

The table:is-selection attribute specifies whether the database range includes a complete database, or a selection of records from a database only.

<define name="table-database-range-attlist" combine="interleave">

<optional>

<attribute name="table:is-selection" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### On Update Keep Styles

The table:on-update-keep-styles attribute specifies the behavior if the database range is updated. If the attribute value is “true”, the cell styles that are assigned to the cells in the first non-label row of the database range are used for all rows with in the database range. If the attribute value is “false”, all cells in the database range get the default cell style of the document assigned.

<define name="table-database-range-attlist" combine="interleave">

<optional>

<attribute name="table:on-update-keep-styles" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### On Update Keep Size

The table:on-update-keep-size attribute specifies the behavior of the database range if the size of the data in the data source changes. If the attribute value is true, the range retains its size. If the attribute value is false, the range does not retain its size.

<define name="table-database-range-attlist" combine="interleave">

<optional>

<attribute name="table:on-update-keep-size" a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Has Persistent Data

The table:has-persistent-data attribute specifies whether the current data in a database range is saved when the document itself is saved.

<define name="table-database-range-attlist" combine="interleave">

<optional>

<attribute name="table:has-persistent-data" a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Orientation

The table:orientation attribute specifies the orientation of the database range. The values of this attribute are row and column. The orientation is for instance used when sorting database ranges (see 8.6.5). If the orientation is row, the sorting takes places for rows, otherwise for columns.

<define name="table-database-range-attlist" combine="interleave">

<optional>

<attribute name="table:orientation" a:defaultValue="row">

<choice>

<value>column</value>

<value>row</value>

</choice>

</attribute>

</optional>

</define>

#### Contains Header

The table:contains-header attribute specifies whether or not the the content of the database range's first row or column should be used to specify labels. If the attribute's value is true, the content of the first cell within a row or column can be used to reference the whole row or column within many spreadsheet operations, for instance from within data pilots.

<define name="table-database-range-attlist" combine="interleave">

<optional>

<attribute name="table:contains-header" a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Display Filter Buttons

The table:display-filter-buttons buttons attribute specifies whether or not to display filter buttons. Filter buttons are list box controls displayed in the label cells whose list entries are the values that exist in the labeled row or column. Selecting one of these entries equals applying a filter to the database range that selects all row or columns where the cells in the labeled row or column have the selected value.

<define name="table-database-range-attlist" combine="interleave">

<optional>

<attribute name="table:display-filter-buttons"

a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Target Range Address

The table:target-range-address attribute specifies the cell range address of the database range. A differentiation between absolute and relative addresses is not possible. Therefore, a table name must be specified in the address and dollar signs are ignored.

<define name="table-database-range-attlist" combine="interleave">

<attribute name="table:target-range-address">

<ref name="cellRangeAddress"/>

</attribute>

</define>

#### Refresh Delay

The table:refresh-delay attribute specifies a time delay between automatic refresh actions.

<define name="table-database-range-attlist" combine="interleave">

<optional>

<attribute name="table:refresh-delay">

<ref name="boolean"/>

</attribute>

</optional>

</define>

### Database Source SQL

The <table:database-source-sql> element describes an SQL database that contains the source data of the database range.

<define name="table-database-source-sql">

<element name="table:database-source-sql">

<ref name="table-database-source-sql-attlist"/>

<empty/>

</element>

</define>

#### Database Name

A table:database-name attribute specifies the name of the SQL database where the data is imported from.

<define name="table-database-source-sql-attlist" combine="interleave">

<attribute name="table:database-name">

<ref name="string"/>

</attribute>

</define>

#### SQL Statement

An table:sql-statement attribute specifies the SQL statement to use when importing data from an SQL database.

<define name="table-database-source-sql-attlist" combine="interleave">

<attribute name="table:sql-statement">

<ref name="string"/>

</attribute>

</define>

#### Parse SQL Statement

A table:parse-sql-statement attribute specifies whether or not the application will parse SQL statements.

<define name="table-database-source-sql-attlist" combine="interleave">

<optional>

<attribute name="table:parse-sql-statement" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

### Database Source Table

The database source table element <table:database-source-table> specifies that the source data of the database range is stored in a database table.

<define name="table-database-source-query">

<element name="table:database-source-table">

<ref name="table-database-source-table-attlist"/>

<empty/>

</element>

</define>

#### Database Name

The table:database-name name attribute specifies the name of the database where the data is imported from.

<define name="table-database-source-table-attlist" combine="interleave">

<attribute name="table:database-name">

<ref name="string"/>

</attribute>

</define>

#### Table Name

A table:database-table-name attribute specifies the database table that data is imported from.

<define name="table-database-source-table-attlist" combine="interleave">

<attribute name="table:database-table-name">

<ref name="string"/>

</attribute>

</define>

### Database Source Query

The database source query element <table:database-source-query> specifies that the source data of the database range is is the result of a database query.

<define name="table-database-source-table">

<element name="table:database-source-query">

<ref name="table-database-source-query-attlist"/>

<empty/>

</element>

</define>

#### Database Name

A table:database-name attribute specifies the name of the database that data is imported from.

<define name="table-database-source-query-attlist" combine="interleave">

<attribute name="table:database-name">

<ref name="string"/>

</attribute>

</define>

#### Query Name

A table:query-name attribute specifies the query to perform on the database whose data is being imported.

<define name="table-database-source-query-attlist" combine="interleave">

<attribute name="table:query-name">

<ref name="string"/>

</attribute>

</define>

### Sort

The sort element <table:sort> describes the sort keys that should be applied to a database range.

<define name="table-sort">

<element name="table:sort">

<ref name="table-sort-attlist"/>

<oneOrMore>

<ref name="table-sort-by"/>

</oneOrMore>

</element>

</define>

#### Bind Styles to Content

The table:bind-styles-to-content attribute specifies whether or not cells retain their style attributes after a sort operation.

<define name="table-sort-attlist" combine="interleave">

<optional>

<attribute name="table:bind-styles-to-content" a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Target Range Address

If the optional table:target-range-address attribute is present, the result of the sort is copied into the specified cell range rather than in the source cell range specified by the database range. A differentiation between absolute and relative addresses is not possible. Therefore, a table name has to exist in the address and dollar signs are ignored.

<define name="table-sort-attlist" combine="interleave">

<optional>

<attribute name="table:target-range-address">

<ref name="cellRangeAddress"/>

</attribute>

</optional>

</define>

#### Case Sensitive

The table:case-sensitive attribute specifies whether or not the sort operation is case sensitive.

<define name="table-sort-attlist" combine="interleave">

<optional>

<attribute name="table:case-sensitive" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Language

The table:language attribute specifies the natural language in which the comparison will occur.

<define name="table-sort-attlist" combine="interleave">

<optional>

<attribute name="table:language">

<ref name="languageCode"/>

</attribute>

</optional>

</define>

#### Country

The table:country attribute specifies the country specific rules to be used in string comparisons for a particular natural language.

<define name="table-sort-attlist" combine="interleave">

<optional>

<attribute name="table:country">

<ref name="countryCode"/>

</attribute>

</optional>

</define>

#### Algorithm

The table:algorithm attribute specifies the algorithm used to compare sort keys. The attribute's value is a an application but also language and country specific sort algorithm name like “phonetic (alphanumeric first)”. To avoid name clashed between different applications, the name should start with a namespace prefix

<define name="table-sort-attlist" combine="interleave">

<optional>

<attribute name="table:algorithm">

<ref name="string"/>

</attribute>

</optional>

</define>

### Sort By

The sort by element <table:sort-by> specifies a key or field to sort, the data type of this field, and how to sort it.

<define name="table-sort-by">

<element name="table:sort-by">

<ref name="table-sort-by-attlist"/>

<empty/>

</element>

</define>

#### Field Number

The table:field-number number attribute specifies the row or column number to sort by. It is the number of a row or column within the database range.

<define name="table-sort-by-attlist" combine="interleave">

<attribute name="table:field-number">

<ref name="nonNegativeInteger"/>

</attribute>

</define>

#### Data Type

The table:data-type attribute specifies the data type of the field to be sorted. Its value can be text, number, automatic or the name of user defined sort order. If the attribute value is automatic, the application must determine what type of data is in the field. User defined sort orders are for instance lists of names of months. Specifying user defined sort orders is application specific.

<define name="table-sort-by-attlist" combine="interleave">

<optional>

<attribute name="table:data-type" a:defaultValue="automatic">

<choice>

<value>text</value>

<value>number</value>

<value>automatic</value>

<ref name="string"/>

</choice>

</attribute>

</optional>

</define>

#### Order

The table:order attribute specifies whether to sort the data in ascending or descending order.

<define name="table-sort-by-attlist" combine="interleave">

<optional>

<attribute name="table:order" a:defaultValue="ascending">

<choice>

<value>ascending</value>

<value>descending</value>

</choice>

</attribute>

</optional>

</define>

### Subtotal Rules

The subtotal rules element <table:subtotal-rules> specifies that provisional results (called subtotals) should be calculated for a database range. It contains information about the row or column provisional results should be calculated for, and also how these results are calculated. To calculate provisional results, the cell values of a row or column a grouped by their value, that is, all cells with the same content in the same field form a group. A provisional result is calculated and displayed at the end of each group.

<define name="table-subtotal-rules">

<element name="table:subtotal-rules">

<ref name="table-subtotal-rules-attlist"/>

<optional>

<ref name="table-sort-groups"/>

</optional>

<zeroOrMore>

<ref name="table-subtotal-rule"/>

</zeroOrMore>

</element>

</define>

#### Bind Styles To Content

The table:bind-styles-to-content attribute specifies whether or not cells retain their style after a subtotal calculation. This attribute is only evaluated if the table:sort-groups element is present.

<define name="table-subtotal-rules-attlist" combine="interleave">

<optional>

<attribute name="table:bind-styles-to-content" a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Case Sensitive

The table:case-sensitive attribute specifies whether or not the case of characters is important when comparing entries, for example, when sorting groups.

<define name="table-subtotal-rules-attlist" combine="interleave">

<optional>

<attribute name="table:case-sensitive" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Page Breaks On Group Change

The table:page-breaks-on-group-change on group change attribute specifies whether or not to insert a page break after the subtotal for each group.

<define name="table-subtotal-rules-attlist" combine="interleave">

<optional>

<attribute name="table:page-breaks-on-group-change"

a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

### Subtotal Sort Groups

The optional sort groups element <table:sort-groups> specifies that columns or rows are sorted before grouping them, and how to sort them. It belongs to the subtotal rules element, see section 8.6.7.

<define name="table-sort-groups">

<element name="table:sort-groups">

<ref name="table-sort-groups-attlist"/>

<empty/>

</element>

</define>

#### Data Type

The table:data-type attribute specifies the data type of the column or row group to sort. See section 8.6.6 for details.

<define name="table-sort-groups-attlist" combine="interleave">

<optional>

<attribute name="table:data-type" a:defaultValue="automatic">

<choice>

<value>text</value>

<value>number</value>

<value>automatic</value>

<ref name="string"/>

</choice>

</attribute>

</optional>

</define>

#### Order

The table:order attribute specifies whether to sort the group data in ascending or descending order. See section 8.6.6 for details.

<define name="table-sort-groups-attlist" combine="interleave">

<optional>

<attribute name="table:order" a:defaultValue="ascending">

<choice>

<value>ascending</value>

<value>descending</value>

</choice>

</attribute>

</optional>

</define>

### Subtotal Rule

The subtotal rule element <table:subtotal-rule> describes how to calculate the subtotals for a certain row or column. The rule contains the group field number, which specifies the column group for which the rule is used, and one or more subtotal fields, which specify a row a column where subtotals should be calculated as well as the function to use for the calculation.

<define name="table-subtotal-rule">

<element name="table:subtotal-rule">

<ref name="table-subtotal-rule-attlist"/>

<zeroOrMore>

<ref name="table-subtotal-field"/>

</zeroOrMore>

</element>

</define>

#### Group By Field Number

The table:group-by-field-number attribute specifies the field, for example, a column, that is to be grouped. It is the number of a row or column within the database range.

<define name="table-subtotal-rule-attlist" combine="interleave">

<attribute name="table:group-by-field-number">

<ref name="nonNegativeInteger"/>

</attribute>

</define>

### Subtotal Field

The subtotal field element <table:subtotal-field> contains the field number and the function that is used to calculate a provisional result.

<define name="table-subtotal-field">

<element name="table:subtotal-field">

<ref name="table-subtotal-field-attlist"/>

<empty/>

</element>

</define>

#### Field Number

The table:field-number attribute specifies the row or column a subtotal should be calculated for. It is the number of a row or column within the database range.

<define name="table-subtotal-field-attlist" combine="interleave">

<attribute name="table:field-number">

<ref name="nonNegativeInteger"/>

</attribute>

</define>

#### Function

The table:function attribute specifies what kind of subtotals to calculate. The following are possible values for this attribute: auto, average, count, countnums, max, min, product, stdev, stdevp, sum, var and varp.

<define name="table-subtotal-field-attlist" combine="interleave">

<attribute name="table:function">

<choice>

<value>auto</value>

<value>average</value>

<value>count</value>

<value>countnums</value>

<value>max</value>

<value>min</value>

<value>product</value>

<value>stdev</value>

<value>stdevp</value>

<value>sum</value>

<value>var</value>

<value>varp</value>

<ref name="string"/>

</choice>

</attribute>

</define>

**Example: Subtotal field**

<table:database-range table:range-position="sampletable.A1:sampletable.G20" table:name="sample">

<table:database-source-table table:database-name="sampleDB" table:table-name="sampleTable"/>

<table:filter ...>

...

</table:filter>

<table:sort>

<table:sort-by table:field-number=1/>

</table:sort>

<table:subtotal-rules>

<table:sort-groups/>

<table:subtotal-rule table:column-group "3">

<table:subtotal-field table:field-number="1"

table:function="sum"/>

</table:subtotal-rule>

</table:subtotal-rules>

</table:database-range>

## Filters

Filters specify that only rows that match certain conditions should be visible

### Table Filter

The table filter element <table:filter> describes how the data contained in a database range or data pilot tables is filtered. The condition specified in the element are applied to all rows specified in the database range or the data pilot table. Rows where the condition does not evaluate to true are made invisible.

<define name="table-filter">

<element name="table:filter">

<ref name="table-filter-attlist"/>

<choice>

<ref name="table-filter-condition"/>

<ref name="table-filter-and"/>

<ref name="table-filter-or"/>

</choice>

</element>

</define>

#### Target Range Address

If the optional table:target-range-address attribute is present, the result of the filter is copied into the specified cell range but all table rows remain visible. If the attribute is not present, the rows that do not match the filter conditions are not displayed. A differentiation between absolute and relative addresses is not possible. Therefore, a table name has to exist in the address and dollar signs are ignored.

<define name="table-filter-attlist" combine="interleave">

<optional>

<attribute name="table:target-range-address">

<ref name="cellRangeAddress"/>

</attribute>

</optional>

</define>

#### Condition Source

The table:condition-source attribute specifies whether the condition is contained in the filter or encoded in a table range. If the value is self the condition is specified by the <table:filter> element's child elements. If the value is cell-range the condition is encoded into the cell range specified by the table:condition-source-range-address attribute.

<define name="table-filter-attlist" combine="interleave">

<optional>

<attribute name="table:condition-source" a:defaultValue="self">

<choice>

<value>self</value>

<value>cell-range</value>

</choice>

</attribute>

</optional>

</define>

#### Condition Source Range Address

The table:condition-source-range-address attribute specifies a cell range that contains encoded conditions. The first row of the cell range has to contain the labels of the columns whose content should be filtered. The following rows contain conditions that have to evaluate to true for the cells contained in the columns. The conditions in each row are connected by an “and” operation, while the rows are connected by an “or” operation. This means that a row is of the source table is displayed if there is at least one row in the condition range where all conditions evaluate to true if they are applied to the columns specified in the first row of the condition range.

Example: If the condition source range is E1:F3 (shown yellow) and the source range is A1:C3 (shown green), only rows 2 and 3 are displayed.

|  | A | B | C | D | E | F | G | G | I |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 3 | 4 |  | A | B |  |  |  |
| 2 | 1 | 5 | 6 |  | =1 | =5 |  |  |  |
| 3 | 2 | 8 | 9 |  | >=2 |  |  |  |  |

Row 2 is displayed because the cell in column A has the value 1 and the cell in column B the value 5, so all conditions of the 2nd row of the condition range evaluate to true. Row 3 is displayed because the cell in column A is larger or equal than 2, and therefor the only condition in the the 3rd row of the condition range evaluates to true.

<define name="table-filter-attlist" combine="interleave">

<optional>

<attribute name="table:condition-source-range-address">

<ref name="cellRangeAddress"/>

</attribute>

</optional>

</define>

#### Display Duplicates

The table:display-duplicates attribute specifies whether or not to display duplicate matches in the result.

<define name="table-filter-attlist" combine="interleave">

<optional>

<attribute name="table:display-duplicates" a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

### Filter And

The <table:filter-and> element specifies that the logical operator AND is applied to the conditions specified by the element's child elements.

<define name="table-filter-and">

<element name="table:filter-and">

<oneOrMore>

<choice>

<ref name="table-filter-or"/>

<ref name="table-filter-condition"/>

</choice>

</oneOrMore>

</element>

</define>

### Filter Or

The <table:filter-or> element specifies that the logical operator OR is applied to the conditions specified by the element's child elements.

<define name="table-filter-or">

<element name="table:filter-or">

<oneOrMore>

<choice>

<ref name="table-filter-and"/>

<ref name="table-filter-condition"/>

</choice>

</oneOrMore>

</element>

</define>

### Filter Condition

The table <table:filter-condition> element describes a single condition to apply in a filter operation.

<define name="table-filter-condition">

<element name="table:filter-condition">

<ref name="table-filter-condition-attlist"/>

<empty/>

</element>

</define>

#### Field Number

The field number attribute table:field-number specifies which field to use for the condition. A field number is the number of a row or column in the source range of the filter.

<define name="table-filter-condition-attlist" combine="interleave">

<attribute name="table:field-number">

<ref name="nonNegativeInteger"/>

</attribute>

</define>

#### Value

The table:value attribute specifies a value for the filter condition.

<define name="table-filter-condition-attlist" combine="interleave">

<attribute name="table:value">

<ref name="string"/>

</attribute>

</define>

#### Operator

The operator attribute table:operator specifies what operator to use in the filter condition. This means that each cell contained in the columns specified by the field number (i.e., the table:field-number attribute) is compared with the value (i.e., the table:value attribute) using the given operator. The result of this comparison is the result of the filter sub-conditions specified by the <table:filter-condition> element.

The operators may or may not make use of regular expressions. The operators that use regular expressions are the following:

* match (matches)
* !match (does not match)

In both case, the table:value attribute contains the regular expression that the table cells have to match or must not match.

The relational operators that do not use regular expressions are:

* = (Equal to)
* != (Not equal to)
* < (Less than)
* > (Greater than)
* <= (Less than or equal to)
* >= (Greater than or equal to)

In addition, operators “empty”, “!empty”, “bottom values”, “top values”, “bottom percent”, and “top percent” can be used. To filter for example the lowest and highest percentage values, the latter two operators can be used.

<define name="table-filter-condition-attlist" combine="interleave">

<attribute name="table:operator">

<ref name="string"/>

</attribute>

</define>

#### Case Sensitive

The table:case-sensitive case sensitive attribute determines whether a filter condition is case sensitive.

<define name="table-filter-condition-attlist" combine="interleave">

<optional>

<attribute name="table:case-sensitive" a:defaultValue="false">

<ref name="string"/>

</attribute>

</optional>

</define>

#### Data Type

The table:data-type attribute specifies whether comparison shall take place as text or as numeric values.

<define name="table-filter-condition-attlist" combine="interleave">

<optional>

<attribute name="table:data-type" a:defaultValue="text">

<choice>

<value>text</value>

<value>number</value>

</choice>

</attribute>

</optional>

</define>

**Example:Representation of a filter**

<filter>

<filter-or>

<filter-and>

<filter-condition table:field-number=1 table:operator="="

table:value="Doe"/>

<filter-condition table:field-number=2 table:operator="="

table:value="John"/>

</filter-and>

<filter-and>

<filter-condition table:field-number=1 table:operator="="

table:value="Burns"/>

<filter-condition table:field-number=2 table:operator="="

table:value="Michael"/>

</filter-and>

</filter-or>

</filter>

## Data Pilot Tables

Data pilot tables allow it to analyze and evaluate data contained in spreadsheet tables. The data pilot tables element <table:data-pilot-tables> contains the collection of all data pilot tables within a document.

<define name="table-data-pilot-tables">

<element name="table:data-pilot-tables">

<zeroOrMore>

<ref name="table-data-pilot-table"/>

</zeroOrMore>

</element>

</define>

### Data Pilot Table

The <table:data-pilot-table> specifies a single data pilot table. Within data pilot tables, all combinations of values that exist in selected columns are collected, and for each of these combinations a formula is applied to the cells of other columns.

Example: Given is the following source table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D |  |
| 1 | **Article** | **City** | **Country** | **Amount** | **Price** |
| 2 | Main Unit | Hamburg | Germany | 1 | 12 |
| 3 | Monitor | Hamburg | Germany | 2 | 15 |
| 4 | Printer | Paris | France | 4 | 13 |
| 5 | Monitor | Paris | France | 2 | 14 |
| 6 | Main Unit | Paris | France | 1 | 12 |
| 7 | Monitor | Hamburg | Germany | 2 | 10 |
| 8 | Printer | Paris | France | 2 | 16 |

The following data pilot table groups the source table by the columns “County”, “City” and “Article” and calculates the sum of the “Amount” as well as of the “Price” columns for each combinations of values of these three columns. The values of the Country and City columns are shown in columns, while the ones of the Article columns are shown in rows.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | Article |  |  |  |
| Country | City | Data | Main Unit | Monitor | Printer | **Total** |
| France | Paris | Sum - Amount | 1 | 2 | 6 | **9** |
|  |  | Sum - Price | 12 | 14 | 29 | **55** |
| Germany | Hamburg | Sum - Amount | 1 | 4 |  | **5** |
|  |  | Sum - Price | 12 | 25 |  | **37** |
| **Total sum - Amount** |  |  | **2** | **6** | **6** | **14** |
| **Total sum - Price** |  |  | **24** | **39** | **29** | **92** |

The columns that are used for grouping (here “County”, “City” and “Article”) are called category columns. The columns for which a formula is calculated based on the value combinations of the category columns (here “Amount” and “Price”) are called data columns. The individual values that exists within a category column are called members.

In general, the behavior of a data pilot is specified by fields, where each field has a name and a so called orientation. The category columns are specified by fields with the orientation “row” or “column” and the data columns are specified by fields that have the orientation “data”. In the above example, “Article” is a field with the orientation column, while “Country” and “City” are fields with the orientation row. “Amount” and “Price” are fields with “data” orientation.

A third kind of fields are data layout fields. Data layout fields are not connected to a column in the source table, but have the only the purpose to change the layout of the data pilot table. In the example, “Data” is a data layout field.

The order in which fields are specified is of relevance. It specified the order in which the data of category columns is grouped and results are displayed. The data pilot table below displays how the data pilot table changes if for instance the data layout field is specified before the category column fields.

Example: A data pilot with a modified layout

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | Article |  |  |  |
| Data | Country | City | Main Unit | Monitor | Printer | **Total** |
| Sum - Amount | France | Paris | 1 | 2 | 6 | **9** |
|  | Germany | Hamburg | 1 | 4 |  | **5** |
| Sum - Price | France | Paris | 12 | 14 | 29 | **55** |
|  | Germany | Hamburg | 12 | 25 |  | **37** |
| **Total sum - Amount** |  |  | **2** | **6** | **6** | **14** |
| **Total sum - Price** |  |  | **24** | **39** | **29** | **92** |

The attributes associated with the data pilot table element are:

* Data pilot table name
* Application data
* Grand total
* Ignore empty rows
* Identify categories
* Target range address
* Show Filter Button
* Drill Down On Double Click

<define name="table-data-pilot-table">

<element name="table:data-pilot-table">

<ref name="table-data-pilot-table-attlist"/>

<optional>

<choice>

<ref name="table-database-source-sql"/>

<ref name="table-database-source-table"/>

<ref name="table-database-source-query"/>

<ref name="table-source-service"/>

<ref name="table-source-cell-range"/>

</choice>

</optional>

<oneOrMore>

<ref name="table-data-pilot-field"/>

</oneOrMore>

</element>

</define>

#### Data Pilot Table Source

The source of the data pilot table is either stored in a database, that is, a database table itself, a SQL query or a named query, or it is a cell range located within the same document. It can also be provided by an external component in an implementation dependent way.

The source of the data pilot table is specified by one of the following elements that are contained in the <table:data-pilot-table> element:

* <table:database-source-sql> (see section 8.6.2)
* <table:database-source-table> (see section 8.6.3)
* <table:database-source-query> (see section 8.6.4)
* <table:source-cell-range> (see section 8.8.2)
* <table:source-service> (see section 8.8.3)

#### Data Pilot Table Name

The table:name attribute specifies the name of the data pilot table.

<define name="table-data-pilot-table-attlist" combine="interleave">

<attribute name="table:name">

<ref name="string"/>

</attribute>

</define>

#### Application Data

The table:application-data attribute specifies extra information about the data pilot table, which can be used by the application, for instance within macros. This data does not influence the behavior of the data pilot.

<define name="table-data-pilot-table-attlist" combine="interleave">

<optional>

<attribute name="table:application-data">

<ref name="string"/>

</attribute>

</optional>

</define>

#### Grand Total

The table:grand-total attribute specifies whether a grand total column, row, or both should be displayed in addition to values calculated for each combination of values in the category columns. In the above example, grand totals are enabled. They are displayed in the row and column labeled “Total”.

<define name="table-data-pilot-table-attlist" combine="interleave">

<optional>

<attribute name="table:grand-total" a:defaultValue="both">

<choice>

<value>none</value>

<value>row</value>

<value>column</value>

<value>both</value>

</choice>

</attribute>

</optional>

</define>

#### Ignore Empty Rows

The table:ignore-empty-rows attribute specifies whether or not empty rows in the source range should be ignored.

<define name="table-data-pilot-table-attlist" combine="interleave">

<optional>

<attribute name="table:ignore-empty-rows" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Identify Categories

The table:identify-categories attribute specifies whether rows that do not contain a value in one of the category columns should use the value of the nearest ancestor row that has a value, or whether such rows should be moved into a group (or category) of its own. If the attribute's value is false, empty values form a category of its own.

<define name="table-data-pilot-table-attlist" combine="interleave">

<optional>

<attribute name="table:identify-categories" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Target Range Address

The table:target-range-address attribute specifies where the target range of the data pilot table output, that is, where the data pilot table is displayed. A differentiation between absolute and relative addresses is not possible, that is, the address is interpreted as an absolute address even if it contains dollar signs. The range address must contain a table name.

<define name="table-data-pilot-table-attlist" combine="interleave">

<attribute name="table:target-range-address">

<ref name="cellRangeAddress"/>

</attribute>

</define>

#### Buttons

Within a data pilot table, some cells might be displayed as buttons to allow interactive operations on the table like changing the order of columns. The table:buttons attribute specifies all cells which should be displayed this way. Its value is a list of cell-addresses. A differentiation between absolute and relative addresses is not possible, that is, the addresses are interpreted as absolute addresses even if they contain dollar signs. All addresses must contain a table name.

In the examples above, button cells are displayed with a gray background.

<define name="table-data-pilot-table-attlist" combine="interleave">

<optional>

<attribute name="table:buttons">

<ref name="cellRangeAddressList"/>

</attribute>

</optional>

</define>

#### Show Filter Button

The table:show-filter-button attribute specifies whether a filter button is shown in the UI within the Data Pilot. A filter button displays a filter dialog if pushed.

<define name="table-data-pilot-table-attlist" combine="interleave">

<optional>

<attribute name="table:show-filter-button" a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Drill Down On Double Click

The table:drill-down-on-double-click attribute specifies how the data pilot table reacts on a double click into the data pilot table. If the attribute's value is false, a double click on a member label or the empty area next to it starts the edit mode of the table cell, like for cells outside of the data pilot table. This can then be used to rename group fields or members. If the attribute's value is true, a double click on a member label or the empty area next to it shows or hides details for that member. A double click elsewhere in a data pilot table has no effect.

<define name="table-data-pilot-table-attlist" combine="interleave">

<optional>

<attribute name="table:drill-down-on-double-click"

a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

### Source Cell Range

If the source of a data pilot table is a cell range, the <table:source-cell-range> element contains information about the cell range and how the data pilot table gets the data from the range. Before the source data is processed by the data pilot data, a filter may be applied to it. This filter has to be specified by a <table:filter> child element.

<define name="table-source-cell-range">

<element name="table:source-cell-range">

<ref name="table-source-cell-range-attlist"/>

<optional>

<ref name="table-filter"/>

</optional>

</element>

</define>

The only attribute that may be associated with the source cell range element is:

* Cell range address

#### Cell Range Address

The table:cell-range-address attribute specifies the cell range containing the source data. The source cell range's address must be absolute. Therefore, the cell range address must contain a table name and dollar signs are ignored.

<define name="table-source-cell-range-attlist" combine="interleave">

<attribute name="table:cell-range-address">

<ref name="cellRangeAddress"/>

</attribute>

</define>

### Source Service

The source of a data pilot table can be “service”, that is, it can be provided by an external component. The source service element <table:source-service> contains information about the service which is used to create the data pilot table.

<define name="table-source-service">

<element name="table:source-service">

<ref name="table-source-service-attlist"/>

<empty/>

</element>

</define>

The attributes that may be associated with this element are:

* Service name
* Source name
* Object name
* Source user name
* Source password

#### Service Name

The table:name attribute specifies the name of the service. The value of this attribute is implementation specific.

<define name="table-source-service-attlist" combine="interleave">

<attribute name="table:name">

<ref name="string"/>

</attribute>

</define>

#### Source Name

The table:source-name attribute specifies a source name that is passed to the service implementation. Its value is application and service specific.

<define name="table-source-service-attlist" combine="interleave">

<attribute name="table:source-name">

<ref name="string"/>

</attribute>

</define>

#### Object Name

The table:object-name attribute specifies the name of the object in the source which contains the data and is passed to the service implementation. Its value is application and service specific.

<define name="table-source-service-attlist" combine="interleave">

<attribute name="table:object-name">

<ref name="string"/>

</attribute>

</define>

#### Source User Name

The table:user-name attribute specifies the user name required to access the source. It is passed to the service implementation. Its value is application and service specific.

<define name="table-source-service-attlist" combine="interleave">

<optional>

<attribute name="table:user-name">

<ref name="string"/>

</attribute>

</optional>

</define>

#### Source Password

The table:password attribute specifies the password required to access the source. It is passed to the service implementation. Its value is application and service specific.

<define name="table-source-service-attlist" combine="interleave">

<optional>

<attribute name="table:password">

<ref name="string"/>

</attribute>

</optional>

</define>

### Data Pilot Field

A data pilot table's fields are specified by <table:data-pilot-field> elements.

<define name="table-data-pilot-field">

<element name="table:data-pilot-field">

<ref name="table-data-pilot-field-attlist"/>

<optional>

<ref name="table-data-pilot-level"/>

</optional>

<optional>

<ref name="table-data-pilot-field-reference"/>

</optional>

<optional>

<ref name="table-data-pilot-groups"/>

</optional>

</element>

</define>

The attributes that may be associated with the data pilot field element are:

* Source field name
* Orientation
* Is data layout field
* Function
* Used hierarchy

#### Source Field Name

For fields that specify category or data columns, the table:source-field-name attribute specifies the name or label of the column the field is connected to. If the source of the data pilot table is for instance a cell range, then the attribute's value has to be the column's label.

There can be multiple <table:data-pilot-field> elements with the same value for this attribute.

<define name="table-data-pilot-field-attlist" combine="interleave">

<attribute name="table:source-field-name">

<ref name="string"/>

</attribute>

</define>

#### Orientation

The table:orientation attribute specifies the orientation of the source field. If the value is data, then the field specifies a data column. If the value is row or column, then the field specifies a category column. The value hidden is used for fields that have a corresponding column in the data pilot's source, but are not visible within the data pilot table. The value page indicates that an automatic filter (i.e., one that allows to choose one of the values that are contained in the column) should be generated for the corresponding column. In this case, an additional field with row, column or data orientation has to exist for the column.

If the attribute value is page, the table:selected-page attribute can be used to specify which value is selected for the filter.

<define name="table-data-pilot-field-attlist" combine="interleave">

<choice>

<attribute name="table:orientation">

<choice>

<value>row</value>

<value>column</value>

<value>data</value>

<value>hidden</value>

</choice>

</attribute>

<group>

<attribute name="table:orientation">

<value>page</value>

</attribute>

<attribute name="table:selected-page">

<ref name="string"/>

</attribute>

</group>

</choice>

</define>

#### Is Data Layout Field

The table:is-data-layout-field attribute specifies whether a field is a data layout field (see section 8.8.1). Data layout fields usually don't have a name.

<define name="table-data-pilot-field-attlist" combine="interleave">

<optional>

<attribute name="table:is-data-layout-field" a:defaultValue="false">

<ref name="string"/>

</attribute>

</optional>

</define>

#### Function

The table:function attribute specifies the function which is applied to the cell values of data columns. It is only evaluated if the value of the table:orientation attribute is data. Possible values for this attribute are: auto, average, count, countnums, max, min, product, stdev, stdevp, sum, var and varp. For category columns the attribute's value auto can be used that specifies that no function is applied at all.

<define name="table-data-pilot-field-attlist" combine="interleave">

<optional>

<attribute name="table:function">

<choice>

<value>auto</value>

<value>average</value>

<value>count</value>

<value>countnums</value>

<value>max</value>

<value>min</value>

<value>product</value>

<value>stdev</value>

<value>stdevp</value>

<value>sum</value>

<value>var</value>

<value>varp</value>

<ref name="string"/>

</choice>

</attribute>

</optional>

</define>

#### Used Hierarchy

If the data pilot source is provided by an external component or service, the data contained within category columns may not only grouped by its value, but it may be further divided into sub-groups or hierarchies. A date value for instance might be grouped by

* “year”, “month” and “day of month”, or
* “year”, “week” and “day of week”

If an external components supports hierarchies, it has to assign unique numbers to it. These numbers can be used in the table:used-hierarchy attribute to select the hierarchy that should be applied to the source field. The value means that no hierarchy should be applied at all.

<define name="table-data-pilot-field-attlist" combine="interleave">

<optional>

<attribute name="table:used-hierarchy" a:defaultValue="-1">

<ref name="integer"/>

</attribute>

</optional>

</define>

### Data Pilot Level

The data pilot level element <table:data-pilot-level> contains additional information about a data pilot field.

<define name="table-data-pilot-level">

<element name="table:data-pilot-level">

<ref name="table-data-pilot-level-attlist"/>

<optional>

<ref name="table-data-pilot-subtotals"/>

</optional>

<optional>

<ref name="table-data-pilot-members"/>

</optional>

<optional>

<ref name="table-data-pilot-display-info"/>

</optional>

<optional>

<ref name="table-data-pilot-sort-info"/>

</optional>

<optional>

<ref name="table-data-pilot-layout-info"/>

</optional>

</element>

</define>

The attribute that may be associated associate with the data pilot level element is:

* Show empty

#### Show Empty

The table:show-empty attribute specifies whether or not fields that don't have any members should be displayed. If this attribute is not present, the application might or might not display such fields.

<define name="table-data-pilot-level-attlist" combine="interleave">

<optional>

<attribute name="table:show-empty">

<ref name="boolean"/>

</attribute>

</optional>

</define>

### Data Pilot Subtotals

The data pilot subtotals element <table:data-pilot-subtotals> contains information about the provisional results that are displayed for every member of a field and the function used to calculate the result. Several provisional results can be calculated simultaneously. If the element is not present, the application might or might not display provisional results.

<define name="table-data-pilot-subtotals">

<element name="table:data-pilot-subtotals">

<zeroOrMore>

<ref name="table-data-pilot-subtotal"/>

</zeroOrMore>

</element>

</define>

### Data Pilot Subtotal

The data pilot subtotal element <table:data-pilot-subtotal> contains information about a single provision result calculation.

<define name="table-data-pilot-subtotal">

<element name="table:data-pilot-subtotal">

<ref name="table-data-pilot-subtotal-attlist"/>

<empty/>

</element>

</define>

The attribute that may be associated associate with the data pilot subtotal element is:

* Function

#### Function

The table:function attribute specifies the function used for the subtotal. Possible functions are auto, average, count, countnums, max, min, product, stdev, stdevp, sum, var and varp.

<define name="table-data-pilot-subtotal-attlist" combine="interleave">

<attribute name="table:function">

<choice>

<value>auto</value>

<value>average</value>

<value>count</value>

<value>countnums</value>

<value>max</value>

<value>min</value>

<value>product</value>

<value>stdev</value>

<value>stdevp</value>

<value>sum</value>

<value>var</value>

<value>varp</value>

<ref name="string"/>

</choice>

</attribute>

</define>

### Data Pilot Members

For category columns, it can be controlled whether certain members themselves or the information displayed for a certain member actually is displayed or not. The <table:data-pilot-members> element contains such information.

<define name="table-data-pilot-members">

<element name="table:data-pilot-members">

<zeroOrMore>

<ref name="table-data-pilot-member"/>

</zeroOrMore>

</element>

</define>

### Data Pilot Member

The data pilot member element <table:data-pilot-member> specifies which information is displayed for a certain member.

<define name="table-data-pilot-member">

<element name="table:data-pilot-member">

<ref name="table-data-pilot-member-attlist"/>

<empty/>

</element>

</define>

The attributes that may be associated with the data pilot member element are:

* Member name
* Display
* Show details

#### Member Name

The table:name attribute specifies the value for which display information is specified.

<define name="table-data-pilot-member-attlist" combine="interleave">

<attribute name="table:name">

<ref name="string"/>

</attribute>

</define>

#### Display

The table:display attribute specifies whether or not a data pilot member is visible at all. If this attribute is not present, the application might or might not display the member.

<define name="table-data-pilot-member-attlist" combine="interleave">

<optional>

<attribute name="table:display">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Show Details

The attribute specifies whether additional fields are displayed for a member. This attribute changes the behavior of a data pilot only if there are several fields with the orientation row or column. If this is the case, and if the attribute's value is false for a field with row or column orientation that is not the last field with this orientation, then no members are displayed for all following fields with the same orientation. Instead of this, the data displayed for these fields will be summarized.

<define name="table-data-pilot-member-attlist" combine="interleave">

<optional>

<attribute name="table:show-details">

<ref name="boolean"/>

</attribute>

</optional>

</define>

### Data Pilot Display Info

The <table:data-pilot-display-info> element restricts the number rows that are displayed for a category field to a specific number of values of a data field.

<define name="table-data-pilot-display-info">

<element name="table:data-pilot-display-info">

<ref name="table-data-pilot-display-info-attlist"/>

<empty/>

</element>

</define>

#### Enabled

The table:enabled attribute specifies whether the <table:data-pilot-display-info> element is evaluated or not.

<define name="table-data-pilot-display-info-attlist" combine="interleave">

<attribute name="table:enabled">

<ref name="boolean"/>

</attribute>

</define>

#### Data Field

The table:data-field attribute specifies the data field whose values are taken into account.

<define name="table-data-pilot-display-info-attlist" combine="interleave">

<attribute name="table:data-field">

<ref name="string"/>

</attribute>

</define>

#### Member Count

The table:member-count attribute specifies how many values from the top or from the bottom of data field's column are shown.

<define name="table-data-pilot-display-info-attlist" combine="interleave">

<attribute name="table:member-count">

<ref name="nonNegativeInteger"/>

</attribute>

</define>

#### Display Member Mode

The table:display-member-mode attribute specifies whether the values specified by table:member-count should be taken from the top or from the bottom of a data field's column.

<define name="table-data-pilot-display-info-attlist" combine="interleave">

<attribute name="table:display-member-mode">

<choice>

<value>from-top</value>

<value>from-bottom</value>

</choice>

</attribute>

</define>

### Data Pilot Sort Info

The <table:data-pilot-sort-info> element specifies how the members of a category field are sorted.

<define name="table-data-pilot-sort-info">

<element name="table:data-pilot-sort-info">

<ref name="table-data-pilot-sort-info-attlist"/>

<empty/>

</element>

</define>

#### Sort Mode

The table:sort-mode attribute describes how to sort the members of a single data pilot field. If the mode is data, then the members of the current category field a sorted according to their values in the data field specified by the table:data-field attribute. If the mode is manual, the user can sort the members in the field manually. If the mode is name, the members in the field are sorted by their name.

<define name="table-data-pilot-sort-info-attlist" combine="interleave">

<choice>

<group>

<attribute name="table:sort-mode">

<value>data</value>

</attribute>

<attribute name="table:data-field">

<ref name="string"/>

</attribute>

</group>

<attribute name="table:sort-mode">

<choice>

<value>none</value>

<value>manual</value>

<value>name</value>

</choice>

</attribute>

</choice>

</define>

#### Sort Order

The table:sort-order attribute specifies whether to sort the members ascending or descending.

<define name="table-data-pilot-sort-info-attlist" combine="interleave">

<attribute name="table:order">

<choice>

<value>ascending</value>

<value>descending</value>

</choice>

</attribute>

</define>

### Data Pilot Layout Info

The <table:data-pilot-layout-info> element describes how to layout the field.

<define name="table-data-pilot-layout-info">

<element name="table:data-pilot-layout-info">

<ref name="table-data-pilot-layout-info-attlist"/>

<empty/>

</element>

</define>

#### Layout Mode

The table:layout-mode attribute describes how to layout the field. It may have the following values:

* tabular-layout: Tabular layout mode is the layout, where each member's name is on the same row as the first member from the following field. Subtotals are always shown below a member's data in this mode.
* outline-subtotals-top: In outline layout mode, the members from the following field start in the row below a member's name, like in traditional database reports. Subtotals are shown at the top (in the same row as the member's name). When the subtotals take up more than one row (manually selected, or because there are several data fields), they are always shown below the member's data, regardless of the setting.
* outline-subtotals-bottom: Like outline-subtotals-top, except that subtotals are shown at the bottom (below the member's data, as in tabular layout mode).

<define name="table-data-pilot-layout-info-attlist" combine="interleave">

<attribute name="table:layout-mode">

<choice>

<value>tabular-layout</value>

<value>outline-subtotals-top</value>

<value>outline-subtotals-bottom</value>

</choice>

</attribute>

</define>

#### Add empty lines

If the attribute table:add-empty-lines has the value true, an empty row is inserted in the data pilot table after the data (including the subtotals) for each member of the field.

<define name="table-data-pilot-layout-info-attlist" combine="interleave">

<attribute name="table:add-empty-lines">

<ref name="boolean"/>

</attribute>

</define>

### Data Pilot Field Reference

The <table:data-pilot-field-reference> element describes data which can be used to modify the displayed values of data fields.

<define name="table-data-pilot-field-reference">

<element name="table:data-pilot-field-reference">

<ref name="table-data-pilot-field-reference-attlist"/>

</element>

</define>

#### Reference Field

The table:field-name attribute references a category field whose members influence the displayed values of the data field the <table:data-pilot-field-reference> is part of.

<define name="table-data-pilot-field-reference-attlist" combine="interleave">

<attribute name="table:field-name">

<ref name="string"/>

</attribute>

</define>

#### Reference Member Type

The table:member-type attribute specifies the member of the referenced category field, whose value within the current data field has to be taken into account. If its value is next (previous) then the value of the data field for the next (previous) visible member of the referenced category field will be taken into account. If its value is named, then the table:member-name specifies the member whose value within the data field is taken into account.

For previous and next, empty members are skipped.

<define name="table-data-pilot-field-reference-attlist" combine="interleave">

<choice>

<group>

<attribute name="table:member-type">

<value>named</value>

</attribute>

<attribute name="table:member-name">

<ref name="string"/>

</attribute>

</group>

<attribute name="table:member-type">

<choice>

<value>previous</value>

<value>next</value>

</choice>

</attribute>

</choice>

</define>

#### Reference Type

The table:type attribute specifies the how the referenced category field influences the displayed values of the data field. It may have one of the following values:

* none: This value means that the results in the data fields are displayed unmodified.
* member-difference: From each result, the value calculated for the category field member specified by the table:member-type and table:member-name attributes is subtracted.
* member-percentage: Each result is divided by the value calculated for the category field member specified by the table:member-type and table:member-name attributes. Division by zero results in an error. Empty results are shown as “0”. If the table:member-type attribute has the value previous, “1” is displayed as first value. If the table:member-type attribute has the value next, “1” is displayed as last value.
* member-percentage-difference: From each result, the value calculated for the category field member specified by the table:member-type and table:member-name attributes is subtracted, and the result is divided by this value again. Division by zero results in an error. Otherwise, the rules for member-difference apply.
* running-total: Each result is added to the sum of the results for preceding members in the referenced category field, in the reference field's sort order, and the total sum is shown.
* row-percentage: Each result is divided by the total result for its row in the data pilot table. If there are several data fields, the total for the result's data field is used. If there are subtotals with manually selected summary functions, the total is calculated with the data field's summary function. Division by zero results in an error.
* column-percentage: Same as row-percentage, but the total for the result's column is used.
* total-percentage: Same as row-percentage, but the grand total for the result's data field is used.
* index: The row and column totals and the grand total are calculated as described above, and then are used to calculate the following expression: (original result \* grand total ) / ( row total \* column total ).Division by zero results in an error.

<define name="table-data-pilot-field-reference-attlist" combine="interleave">

<attribute name="table:type">

<choice>

<value>none</value>

<value>member-difference</value>

<value>member-percentage</value>

<value>member-percentage-difference</value>

<value>running-total</value>

<value>row-percentage</value>

<value>column-percentage</value>

<value>total-percentage</value>

<value>index</value>

</choice>

</attribute>

</define>

### Data Pilot Groups

The <table:data-pilot-groups> element specifies that a data pilot field is a group field. A group field allows grouping of other fields. For example, if a data pilot table contains a column field with the name “city” which has the members “Berlin”, “Munich”, “Frankfurt”, “Hamburg”, “London”, “Manchester”, “Hastings” and “Liverpool”, then one may want to group the cities by their countries. To do so, a group field with name “city2” could be added to the data pilot table, that contains two groups called “England” and “Germany”. Each group here contains a list of the names of its members. In this example, the group “England” would contain “London”, “Manchester”, “Hastings” and “Liverpool”. The group “Germany” would contain “Berlin”, “Munich”, “Frankfurt” and “Hamburg”.

Grouping may also take place for numeric or date values.

<define name="table-data-pilot-groups">

<element name="table:data-pilot-groups">

<ref name="table-data-pilot-groups-attlist"/>

<oneOrMore>

<ref name="table-data-pilot-group"/>

</oneOrMore>

</element>

</define>

#### Source Field Name

The table:source-field-name attribute references the field containing the data that is grouped, if this data differs from the data that is referenced by the field itself.

<define name="table-data-pilot-groups-attlist" combine="interleave">

<attribute name="table:source-field-name">

<ref name="string"/>

</attribute>

</define>

#### Start

If numeric or date values are grouped, the table:date-start and table:start attributes specify the start value for the grouping. All values that are lower than the start value are contained in a single group, while values that are equal to or higher than the start value are grouped as specified by the table:grouped-by and table:step attributes.

If the attribute's value is auto, the lowest value of the field is taken as start value.

<define name="table-data-pilot-groups-attlist" combine="interleave">

<choice>

<attribute name="table:date-start">

<choice>

<ref name="dateOrDateTime"/>

<value>auto</value>

</choice>

</attribute>

<attribute name="table:start">

<choice>

<ref name="double"/>

<value>auto</value>

</choice>

</attribute>

</choice>

</define>

#### End

If numeric or date values are grouped, the table:date-end and table:end attributes specify the end value for the grouping. All values that are higher than the end value are contained in a single group, while values that are equal to or lower than the end value are grouped as specified by the table:grouped-by and table:step attributes.

If the attribute's value is auto, the highest value of the field is taken as end value.

<define name="table-data-pilot-groups-attlist" combine="interleave">

<choice>

<attribute name="table:date-end">

<choice>

<ref name="dateOrDateTime"/>

<value>auto</value>

</choice>

</attribute>

<attribute name="table:end">

<choice>

<ref name="double"/>

<value>auto</value>

</choice>

</attribute>

</choice>

</define>

#### Step

The table:step attribute specifies the grouping of numeric values, by specifying the distance between the groups. For example, if the table:start attribute for the grouping has the value 5, and the table:step attribute has the value 2, all values that are equal to or higher than 5, but also lower than 7 are in one group. All values that are equal to or higher than 7, but also lower than 9 are in next group, and so on, until the end value is reached.

<define name="table-data-pilot-groups-attlist" combine="interleave">

<attribute name="table:step">

<ref name="double"/>

</attribute>

</define>

#### Grouped By

The table:grouped-by attribute specifies the grouping of the date values. Date values can be grouped by seconds, minutes, hours, days, months, quarters or years. It date values are for instance grouped by minutes, all dates or times that are within the same minute are within one group. That, is if the dates 2004-08-27T12:34:46, 2004-08-27T12:34:56 and 2004-08-27T12:35:46 are given, the first two would be within one group, while the last date would be a group of its own.

<define name="table-data-pilot-groups-attlist" combine="interleave">

<attribute name="table:grouped-by">

<choice>

<value>seconds</value>

<value>minutes</value>

<value>hours</value>

<value>days</value>

<value>months</value>

<value>quarters</value>

<value>years</value>

</choice>

</attribute>

</define>

### Data Pilot Group

If grouping takes place by specifying the member names, then the <table:data-pilot-group> element specifies the member names of a single group.

<define name="table-data-pilot-group">

<element name="table:data-pilot-group">

<ref name="table-data-pilot-group-attlist"/>

<oneOrMore>

<ref name="table-data-pilot-group-member"/>

</oneOrMore>

</element>

</define>

#### Name

The table:name attribute specifies the name of the group.

<define name="table-data-pilot-group-attlist" combine="interleave">

<attribute name="table:name">

<ref name="string"/>

</attribute>

</define>

### Data Pilot Group Member

The <table:data-pilot-group-member> element specifies the name of a single group member.

<define name="table-data-pilot-group-member">

<element name="table:data-pilot-group-member">

<ref name="table-data-pilot-group-member-attlist"/>

</element>

</define>

#### Name

The table:name attribute specifies the name of the member.

<define name="table-data-pilot-group-member-attlist" combine="interleave">

<attribute name="table:name">

<ref name="string"/>

</attribute>

</define>

## Consolidation

A consolidation combines data from several independent table ranges. A new table range is calculated by applying a mathematical function to all cells in the source table ranges that have the same relative address within these ranges. A consolidation is defined by the <table:consolidation> element.

<define name="table-consolidation">

<element name="table:consolidation">

<ref name="table-consolidation-attlist"/>

<empty/>

</element>

</define>

The attributes that may be associated with this element are:

* Function
* Source cell range addresses
* Target cell address
* Use label
* Link to source data

#### Function

The table:function attribute contains the function which is used to consolidate the data. Possible functions are auto, average, count, countnums, max, min, product, stdev, stdevp, sum, var and varp.

<define name="table-consolidation-attlist" combine="interleave">

<attribute name="table:function">

<choice>

<value>auto</value>

<value>average</value>

<value>count</value>

<value>countnums</value>

<value>max</value>

<value>min</value>

<value>product</value>

<value>stdev</value>

<value>stdevp</value>

<value>sum</value>

<value>var</value>

<value>varp</value>

<ref name="string"/>

</choice>

</attribute>

</define>

#### Source Cell Range Addresses

The table:source-cell-range-addresses attribute contains a list of cell range addresses that specify the source cell ranges.

<define name="table-consolidation-attlist" combine="interleave">

<attribute name="table:source-cell-range-addresses">

<ref name="cellRangeAddressList"/>

</attribute>

</define>

#### Target Cell Address

The table:target-cell-address attribute contains the target cell address.

<define name="table-consolidation-attlist" combine="interleave">

<attribute name="table:target-cell-address">

<ref name="cellAddress"/>

</attribute>

</define>

#### Use Label

The table:use-label attribute specifies whether or not labels should be used by the consolidation for rows, columns or both. Possible values are none, column, row and both. If labels are used for rows or columns, the mathematical functions is applied to cells with equally labeled rows or columns rather than to cells with the same relative cell address.

<define name="table-consolidation-attlist" combine="interleave">

<optional>

<attribute name="table:use-labels" a:defaultValue="none">

<choice>

<value>none</value>

<value>row</value>

<value>column</value>

<value>both</value>

</choice>

</attribute>

</optional>

</define>

#### Link to Source Data

The table:link-to-source-data attribute specifies whether the data in the consolidation table range should be linked to the source data, so that it is automatically updated if any changes are made to the source data.

<define name="table-consolidation-attlist" combine="interleave">

<optional>

<attribute name="table:link-to-source-data" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

## DDE Links

The <table:dde-links> container element stores all DDE links within a spreadsheet document. Every link contains the DDE Source and the data of the last connection. See section 12.6.3 for details.

See section 12.6 for the use of DDE connections.

<define name="table-dde-links">

<element name="table:dde-links">

<oneOrMore>

<ref name="table-dde-link"/>

</oneOrMore>

</element>

</define>

## Change Tracking in Spreadsheets

Within spreadsheet documents, changes to tables can be tracked. This section describes how this change tracking information is represented.

Change tracking of tables is not supported for text documents.

### Tracked Changes

All changes that have been applied to a spreadsheet document are stored in a list. The list contains an element for each change made to the document. To track the changes to a spreadsheet document, the <table:tracked-changes> element must be present.

<define name="table-tracked-changes">

<element name="table:tracked-changes">

<ref name="table-tracked-changes-attlist"/>

<zeroOrMore>

<choice>

<ref name="table-cell-content-change"/>

<ref name="table-insertion"/>

<ref name="table-deletion"/>

<ref name="table-movement"/>

</choice>

</zeroOrMore>

</element>

</define>

#### Track Changes

The table:track-changes attribute specifies whether or not the change tracking is enabled.

<define name="table-tracked-changes-attlist" combine="interleave">

<optional>

<attribute name="table:track-changes" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

### Insertion

The <table:insertion> element contains the information that is required to identify any insertion of content. This content can be one or more rows, one or more columns, or a table.

<define name="table-insertion">

<element name="table:insertion">

<ref name="table-insertion-attlist"/>

<ref name="common-table-change-attlist"/>

<ref name="office-change-info"/>

<optional>

<ref name="table-dependencies"/>

</optional>

<optional>

<ref name="table-deletions"/>

</optional>

</element>

</define>

The attributes that may be associated with this element are:

* ID (see section 8.11.18)
* Acceptance State (see section 8.11.18)
* Rejecting Change ID (see section 8.11.18)
* Type
* Position
* Count
* Table

#### Type

The table:type attribute specifies the type of the insertion. It can be row, column or table.

<define name="table-insertion-attlist" combine="interleave">

<attribute name="table:type">

<choice>

<value>row</value>

<value>column</value>

<value>table</value>

</choice>

</attribute>

</define>

#### Position

The table:position attribute specifies the position where the insertion was made in the table. Depending on the insertion type, It is either the number of a row, a column or a table.

<define name="table-insertion-attlist" combine="interleave">

<attribute name="table:position">

<ref name="integer"/>

</attribute>

</define>

#### Count

The table:count attribute specifies the count of inserted rows, columns or tables.

<define name="table-insertion-attlist" combine="interleave">

<optional>

<attribute name="table:count" a:defaultValue="1">

<ref name="positiveInteger"/>

</attribute>

</optional>

</define>

#### Table

The table:table attribute specifies the number of the table where the insertion took place. This attribute only exists for column and row insertions.

<define name="table-insertion-attlist" combine="interleave">

<optional>

<attribute name="table:table">

<ref name="integer"/>

</attribute>

</optional>

</define>

Example: Insertion of text in a cell

<table:tracked-changes>

<table:insertion table:id="c001" table:acceptance-state="pending"

table:type="column" table:position="5">

<office:change-info>

<dc:creator>Sascha Ballach</dc:creator>

<dc:date>1999-55-18T12:56:04</dc:date>

</office:change-info>

</table:insertion>

</table:tracked-changes>

### Dependencies

The <table:dependencies> element contains the information on which other tracked changes a tracked change depends. Every element of the tracked-changes can contain a <table:dependencies> element.

<define name="table-dependencies">

<element name="table:dependencies">

<oneOrMore>

<ref name="table-dependency"/>

</oneOrMore>

</element>

</define>

### Dependence

The <table:dependency> element contains the information about one change action on which the parent element depends. The change action on which the current depends is referenced by an id.

<define name="table-dependency">

<element name="table:dependency">

<attribute name="table:id">

<ref name="string"/>

</attribute>

<empty/>

</element>

</define>

### Deletions

The <table:deletions> element contains all deletions which are performed while tracking a single change to a table.

<define name="table-deletions">

<element name="table:deletions">

<oneOrMore>

<choice>

<ref name="table-cell-content-deletion"/>

<ref name="table-change-deletion"/>

</choice>

</oneOrMore>

</element>

</define>

### Cell Content Deletion

The <table:cell-content-deletion> element specifies that a cell content has been deleted. It contains the address of the effected cell and its former content. If a text:id attribute is present, it specifies the id of a previously tracked change for the cell that gets deleted by the current change.

<define name="table-cell-content-deletion">

<element name="table:cell-content-deletion">

<optional>

<attribute name="table:id">

<ref name="string"/>

</attribute>

</optional>

<optional>

<ref name="table-cell-address"/>

</optional>

<optional>

<ref name="table-change-track-table-cell"/>

</optional>

</element>

</define>

### Change Deletion

The <table:change-deletion> element specified the id of a previously tracked change that gets deleted by the current change.

<define name="table-change-deletion">

<element name="table:change-deletion">

<optional>

<attribute name="table:id">

<ref name="string"/>

</attribute>

</optional>

<empty/>

</element>

</define>

### Deletion

A <table:deletion> element contains content that was deleted while change tracking was enabled. The content of a cell that was deleted is either contained in the <table:dependencies>, or in the <table:deletions> element.

<define name="table-deletion">

<element name="table:deletion">

<ref name="table-deletion-attlist"/>

<ref name="common-table-change-attlist"/>

<ref name="office-change-info"/>

<optional>

<ref name="table-dependencies"/>

</optional>

<optional>

<ref name="table-deletions"/>

</optional>

<optional>

<ref name="table-cut-offs"/>

</optional>

</element>

</define>

The attributes that may be associated with this element are:

* ID (see section 8.11.18)
* Acceptance State (see section 8.11.18)
* Rejecting Change ID (see section 8.11.18)
* Type
* Position
* Table
* Multi Deletion Spanned

#### Type

The table:type attribute specifies the type of the deletion. It can be row, column or table.

<define name="table-deletion-attlist" combine="interleave">

<attribute name="table:type">

<choice>

<value>row</value>

<value>column</value>

<value>table</value>

</choice>

</attribute>

</define>

#### Position

The table:position attribute specifies the position where the deletion was made in the table. Depending on the deletion type, It is either the number of a row, a column or a table.

<define name="table-deletion-attlist" combine="interleave">

<attribute name="table:position">

<ref name="integer"/>

</attribute>

</define>

#### Table

The table:table attribute specifies the number of the table where the deletion took place. This attribute only exists for column and row deletions.

<define name="table-deletion-attlist" combine="interleave">

<optional>

<attribute name="table:table">

<ref name="integer"/>

</attribute>

</optional>

</define>

#### Multi Deletion Spanned

If multiple columns or rows were deleted simultaneously, each deleted row or column gets its own <table:deletion> element. The element of the first deleted row or column in this case has to carry a table:multi-deletion-spanned attribute that specifies the total number of deleted rows or columns.

<define name="table-deletion-attlist" combine="interleave">

<optional>

<attribute name="table:multi-deletion-spanned">

<ref name="integer"/>

</attribute>

</optional>

</define>

### Cut Offs

A <table:cut-offs> element contains information about previously tracked insertions or movements where parts of the new content created by this operation now gets deleted. An example for this might be a cell range that has previously been moved and that now overlaps with a row that gets deleted.

<define name="table-cut-offs">

<element name="table:cut-offs">

<choice>

<oneOrMore>

<ref name="table-movement-cut-off"/>

</oneOrMore>

<group>

<ref name="table-insertion-cut-off"/>

<zeroOrMore>

<ref name="table-movement-cut-off"/>

</zeroOrMore>

</group>

</choice>

</element>

</define>

### Insertion Cut Off

The <table:insertion-cut-off> element contains the information where a insertion was deleted and which.

<define name="table-insertion-cut-off">

<element name="table:insertion-cut-off">

<ref name="table-insertion-cut-off-attlist"/>

<empty/>

</element>

</define>

The attributes that may be associated with this element are:

* ID (see section 8.11.18)
* position

#### Id

The table:id attribute contains the id of the insertion where parts of now get deleted.

<define name="table-insertion-cut-off-attlist" combine="interleave">

<attribute name="table:id">

<ref name="string"/>

</attribute>

</define>

#### Position

The table:position attribute specifies the number of the row or column within the insertion that gets deleted.

<define name="table-insertion-cut-off-attlist" combine="interleave">

<attribute name="table:position">

<ref name="integer"/>

</attribute>

</define>

### Movement Cut Off

The <table:movement-cut-off> element contains the information where a movement was deleted and which.

<define name="table-movement-cut-off">

<element name="table:movement-cut-off">

<ref name="table-movement-cut-off-attlist"/>

<empty/>

</element>

</define>

The attributes that may be associated with this element are:

* ID (see section 8.11.18)
* start position, end position, position

#### Start Position, End Position, Position

The table:start-position, table:end-position and table:position attributes specify the position within the movement that gets deleted. If a single row or column gets deleted, the table:position attribute contains its number. If multiple rows or columns get deleted, the table:start-position and table:end-position attributes contain the number of the first (inclusive) and last (exclusive) deleted rows or columns.

<define name="table-movement-cut-off-attlist" combine="interleave">

<choice>

<attribute name="table:position">

<ref name="integer"/>

</attribute>

<group>

<attribute name="table:start-position">

<ref name="integer"/>

</attribute>

<attribute name="table:end-position">

<ref name="integer"/>

</attribute>

</group>

</choice>

</define>

Example: Deletion of a column which do not contain content

<table:tracked-changes>

<table:deletion table:id="c002" table:acceptance-state="pending"

table:type="column" table:position="9">

<office:change-info>

<dc:creator>Sascha Ballach</dc:creator>

<dc:date>1999-05-18T12:56:04</dc:creator>

</office:change-info>

</table:deletion>

</table:tracked-changes>

### Movement

A <table:movement> element contains the information that is required to identify any movement of content. This content can be a cell content or a cell range content.

<define name="table-movement">

<element name="table:movement">

<ref name="common-table-change-attlist"/>

<ref name="table-source-range-address"/>

<ref name="table-target-range-address"/>

<ref name="office-change-info"/>

<optional>

<ref name="table-dependencies"/>

</optional>

<optional>

<ref name="table-deletions"/>

</optional>

</element>

</define>

The attributes that may be associated with this element are:

* ID (see section 8.11.18)
* Acceptance State (see section 8.11.18)
* Rejecting Change ID (see section 8.11.18)

### Target Range Address, Source Range Address

The <table:source-range-address> and <table:target-range-address> specify the source and target cell address or cell range address of a movement.

<define name="table-source-range-address">

<element name="table:source-range-address">

<ref name="common-table-range-attlist"/>

<empty/>

</element>

</define>

<define name="table-target-range-address">

<element name="table:target-range-address">

<ref name="common-table-range-attlist"/>

<empty/>

</element>

</define>

<define name="common-table-range-attlist" combine="interleave">

<choice>

<group>

<ref name="common-table-cell-address-attlist"/>

</group>

<group>

<ref name="common-table-cell-range-address-attlist"/>

</group>

</choice>

</define>

The attributes that may be associated with these elements are either

* Column, Row, and Table, or
* Start column, End column, Start row, End row, Start table, and End table

#### Column, Row, and Table

If the range address is a cell address then the three attributes table:column, table:row and  
table:table specify the column, row and table number of the cell.

<define name="common-table-cell-address-attlist" combine="interleave">

<attribute name="table:column">

<ref name="integer"/>

</attribute>

<attribute name="table:row">

<ref name="integer"/>

</attribute>

<attribute name="table:table">

<ref name="integer"/>

</attribute>

</define>

#### Start Column, End Column, Start Row, End Row, Start Table, and End Table

If the range address is a cell range address instead of a cell address, the attributes table:start-column, table:end-column, table:start-row, table:end-row, table:start-table and table:end-table specify the start and end columns, rows and tables of the range. Start and end numbers both are inclusive.

<define name="common-table-cell-range-address-attlist" combine="interleave">

<attribute name="table:start-column">

<ref name="integer"/>

</attribute>

<attribute name="table:start-row">

<ref name="integer"/>

</attribute>

<attribute name="table:start-table">

<ref name="integer"/>

</attribute>

<attribute name="table:end-column">

<ref name="integer"/>

</attribute>

<attribute name="table:end-row">

<ref name="integer"/>

</attribute>

<attribute name="table:end-table">

<ref name="integer"/>

</attribute>

</define>

Example: Moving a cell

<table:tracked-changes>

<table:movement table:id="ct1">

<table:source-range-address table:column="0" table:row="0"

table:table="0"/>

<table:target-range-address table:column="1" table:row="1"

table:table="0"/>

<office:change-info>

<dc:creator>Michael Brauer</dc:creator>

<dc:date>2003-12-29T11:46:13,21"</dc:date>

</office:change-info>

</table:movement>

</table:tracked-changes>

### Change Track Cell

The <table:change-track-table-cell> element contains all information of a table cell which are needed inside the change tracking elements. The element is very similar to a <table:table-cell> element, but contains some additional information.

<define name="table-change-track-table-cell" combine="interleave">

<element name="table:change-track-table-cell">

<ref name="table-change-track-table-cell-attlist"/>

<zeroOrMore>

<ref name="text-p"/>

</zeroOrMore>

</element>

</define>

#### Cell Address

If the cell is a formula cell, the table:cell-address attribute is required and specifies the original address of the cell used in calculations.

<define name="table-change-track-table-cell-attlist" combine="interleave">

<optional>

<attribute name="table:cell-address">

<ref name="cellAddress"/>

</attribute>

</optional>

</define>

#### Matrix Covered

If the cell is a matrix cell and not the base of the matrix the, table:matrix-covered attribute is necessary and its value has to be true to indicate that the cell is contained in a matrix.

<define name="table-change-track-table-cell-attlist" combine="interleave">

<optional>

<attribute name="table:matrix-covered" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Formulas and Values

The change track table cells additionally supports the attributes table:formula, table:number-matrix-rows-spanned, table:number-matrix-columns-spanned, office:value-type, office:value, office:date-value, office:time-value and office:string-value as described in section 8.1.3.

<define name="table-change-track-table-cell-attlist" combine="interleave">

<optional>

<attribute name="table:formula">

<ref name="string"/>

</attribute>

</optional>

<optional>

<attribute name="table:number-matrix-columns-spanned">

<ref name="positiveInteger"/>

</attribute>

</optional>

<optional>

<attribute name="table:number-matrix-rows-spanned">

<ref name="positiveInteger"/>

</attribute>

</optional>

<optional>

<ref name="common-value-and-type-attlist"/>

</optional>

</define>

### Cell Content Change

A <table:cell-content-change> element contains the information that is required to identify changes of the cell content.

<define name="table-cell-content-change">

<element name="table:cell-content-change">

<ref name="common-table-change-attlist"/>

<ref name="table-cell-address"/>

<ref name="office-change-info"/>

<optional>

<ref name="table-dependencies"/>

</optional>

<optional>

<ref name="table-deletions"/>

</optional>

<ref name="table-previous"/>

</element>

</define>

The attributes that may be associated with this element are:

* ID (see section 8.11.18)
* Acceptance State (see section 8.11.18)
* Rejecting Change ID (see section 8.11.18)

### Cell Address

The <table:cell-address> element contains the address of cell that is changed. Unlike other cell addresses, the address consists of the row, column and table number of the cell. This allows specifying addresses that are outside the valid cell address range, for instance have a negative column number.

<define name="table-cell-address">

<element name="table:cell-address">

<ref name="common-table-cell-address-attlist"/>

<empty/>

</element>

</define>

The attributes that may be associated with this element are:

* Column, Row, and Table number (see section 8.11.13)

### Previous

The table:previous element contains the previous cell content which is overwritten by the current change. If a text:id attribute is present, it specifies the id of a previously tracked change for the cell that gets changed again by the current change.

<define name="table-previous">

<element name="table:previous">

<optional>

<attribute name="table:id">

<ref name="string"/>

</attribute>

</optional>

<ref name="table-change-track-table-cell"/>

</element>

</define>

### Common Change Tracking Attributes

#### Id

The table:id attribute specifies the id of the tracked change.

<define name="common-table-change-attlist" combine="interleave">

<attribute name="table:id">

<ref name="string"/>

</attribute>

</define>

#### Acceptance state

The table:acceptance-state attribute specifies whether the tracked change has been accepted or rejected already, or whether an acceptance or rejection is still pending.

<define name="common-table-change-attlist" combine="interleave">

<optional>

<attribute name="table:acceptance-state" a:defaultValue="pending">

<choice>

<value>accepted</value>

<value>rejected</value>

<value>pending</value>

</choice>

</attribute>

</optional>

</define>

#### Rejecting Change Id

If the table:rejecting-change-id attribute is present, then the current change has been made to the table to implement the rejection of another previously tracked change. The attribute's value is the id of this previously tracked change that has been rejected.

<define name="common-table-change-attlist" combine="interleave">

<optional>

<attribute name="table:rejecting-change-id">

<ref name="string"/>

</attribute>

</optional>

</define>

1. Graphic Content

This chapter provides the specification for the core elements of graphic applications like drawing or presentation applications, and for graphical objects contained in non-graphical applications, like word processor or spreadsheet applications.

## Enhanced Page Features for Graphical Applications

### Handout Master

For applications that support printing handout pages, this element is a template for automatically generating the handout pages. The element <style:handout-master> can contain any types of shapes. The most useful shape is the <draw:page-thumbnail>, which is replaced by actual pages from the document. The <style:handout-master> element is contained in the <office:master-styles> element. The <office:master-styles> must not contain more than one <style:handout-master> element.

<define name="style-handout-master">

<element name="style:handout-master">

<ref name="common-presentation-header-footer-attlist"/>

<ref name="style-handout-master-attlist"/>

<zeroOrMore>

<ref name="shape"/>

</zeroOrMore>

</element>

</define>

The attributes that may be associated with the <style:handout-master> element are:

* Presentation Page Layout (placeholder objects)
* Page Layout (page size, margins etc.)
* Page Style
* Header Declaration
* Footer Declaration
* Date and Time Declaration

#### Presentation Page Layout

The attribute presentation:presentation-page-layout-name links to a <style:presentation-page-layout> element. See section 14.15 for information on the presentation page layout element. This attribute is optional.

<define name="style-handout-master-attlist" combine="interleave">

<optional>

<attribute name="presentation:presentation-page-layout-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

</define>

#### Page Layout

The style:page-layout-name attribute specifies a page layout which contains the sizes, border and orientation of the handout master page. See section 14.3 for details on page layouts.

<define name="style-handout-master-attlist" combine="interleave">

<attribute name="style:page-layout-name">

<ref name="styleNameRef"/>

</attribute>

</define>

#### Page Style

The attribute draw:style-name assigns an additional formatting attributes to a handout master page by assigning a drawing page style. This attribute is optional. The fixed family for page styles is drawing-page.

<define name="style-handout-master-attlist" combine="interleave">

<optional>

<attribute name="draw:style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

</define>

#### Header Declaration

The presentation:use-header-name attribute specifies the name of the header field declaration (see section 9.11.2) that is used for all header fields (see section 9.10.1) that are displayed on the handout master page. See also section 9.1.4.

#### Footer Declaration

The presentation:use-footer-name attribute specifies the name of the footer field declaration (see section 9.11.3) that is used for all footer fields (see section 9.10.2) that are displayed on the handout master page. See also section 9.1.4.

#### Date and Time Declaration

The presentation:use-date-time-name attribute specifies the name of the date-time field declaration (see section 9.11.4) that is used for all date-time fields (see section 9.10.3) that are displayed on the handout master page. See also section 9.1.4.

### Layer Sets

The element <draw:layer-set> may be contained in the master styles of graphical applications. It defines a set of layers. Layers group drawing objects. Drawing objects may be assigned to these layers with the help of their draw:layer-name attribute.

<define name="draw-layer-set">

<element name="draw:layer-set">

<zeroOrMore>

<ref name="draw-layer"/>

</zeroOrMore>

</element>

</define>

### Layer

The <draw:layer> element defines a single layer.

<define name="draw-layer">

<element name="draw:layer">

<ref name="draw-layer-attlist"/>

<optional>

<ref name="svg-title"/>

</optional>

<optional>

<ref name="svg-desc"/>

</optional>

</element>

</define>

The <draw:layer element> may contain the following elements:

* Title (short accessible name). Use the <svg:title> child element as described in section 9.2.20.
* Long description (in support of accessibility). Use the <svg:desc> child element as described in section 9.2.20.

#### Name

Each element <draw:layer> is defined and referenced by its name that is contained in the draw:name attribute . Each drawing object inside a drawing or presentation document can be assigned to a layer. Layers virtually group the object. Each object that is assigned to a layer inherits the settings of the layer.

<define name="draw-layer-attlist" combine="interleave">

<attribute name="draw:name">

<ref name="string"/>

</attribute>

</define>

#### Protection

The draw:protected attribute specifies whether the drawing objects contain in the layer are protected from being modified.

<define name="draw-layer-attlist" combine="interleave">

<optional>

<attribute name="draw:protected" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Display

The draw:display attribute specifies whether the drawing objects contain in the layer are visible on the screen and/or printed.

<define name="draw-layer-attlist" combine="interleave">

<optional>

<attribute name="draw:display" a:defaultValue="always">

<choice>

<value>always</value>

<value>screen</value>

<value>printer</value>

<value>none</value>

</choice>

</attribute>

</optional>

</define>

### Drawing Pages

The element <draw:page> is a container for content in a drawing or presentation document. Drawing pages are used for the following:

* Forms (see section 11.1)
* Drawings (see section 9.2)
* Frames (see section 9.3)
* Presentation Animations (see section 9.7)
* Presentation Notes (see section 9.1.5)

A master page must be assigned to each drawing page.

<define name="draw-page">

<element name="draw:page">

<ref name="common-presentation-header-footer-attlist"/>

<ref name="draw-page-attlist"/>

<optional>

<ref name="office-forms"/>

</optional>

<zeroOrMore>

<ref name="shape"/>

</zeroOrMore>

<optional>

<choice>

<ref name="presentation-animations"/>

<ref name="animation-element"/>

</choice>

</optional>

<optional>

<ref name="presentation-notes"/>

</optional>

</element>

</define>

The attributes that may be associated with the <draw:page> element are:

* Page name
* Page style
* Master page
* Presentation page layout
* Header declaration
* Footer declaration
* Date and time declaration
* ID

The elements that my be included in the <draw:page> element are:

* Forms
* Shapes
* Animations
* Presentation notes

#### Page Name

The draw:name attribute specifies the name of a drawing page. This attribute is optional; if it is used, the name must be unique. If it is not used, the application may generate a unique name.

<define name="draw-page-attlist" combine="interleave">

<optional>

<attribute name="draw:name">

<ref name="string"/>

</attribute>

</optional>

</define>

#### Page Style

The attribute draw:style-name assigns an additional formatting attributes to a drawing page by assigning a drawing page style. This attribute is optional. The fixed family for page styles is drawing-page.

For pages inside a presentation document, attributes from Presentation Page Attributes can also be used.

<define name="draw-page-attlist" combine="interleave">

<optional>

<attribute name="draw:style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

</define>

#### Master Page

Each drawing page must have one master page assigned to it. The master page:

* Defines properties such as the size and borders of the drawing page
* Serves as a container for shapes that are used as a common background

The draw:master-page-name attribute specifies the name of the master page assigned to the drawing page. This attribute is required.

<define name="draw-page-attlist" combine="interleave">

<attribute name="draw:master-page-name">

<ref name="styleNameRef"/>

</attribute>

</define>

#### Presentation Page Layout

If the drawing page was created using a presentation page layout, the attribute presentation:presentation-page-layout-name links to the corresponding <style:presentation-page-layout> element. See section 14.15 for information on the presentation page layout element. This attribute is optional.

<define name="draw-page-attlist" combine="interleave">

<optional>

<attribute name="presentation:presentation-page-layout-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

</define>

#### Header Declaration

The presentation:use-header-name attribute specifies the name of the header field declaration (see section 9.11.2) that is used for all header fields (see section 9.10.1) that are displayed on the page.

<define name="common-presentation-header-footer-attlist" combine="interleave">

<optional>

<attribute name="presentation:use-header-name">

<ref name="string"/>

</attribute>

</optional>

</define>

#### Footer Declaration

The presentation:use-footer-name attribute specifies the name of the footer field declaration (see section 9.11.3) that is used for all footer fields (see section 9.10.2) that are displayed on the page.

<define name="common-presentation-header-footer-attlist" combine="interleave">

<optional>

<attribute name="presentation:use-footer-name">

<ref name="string"/>

</attribute>

</optional>

</define>

#### Date and Time Declaration

The presentation:use-date-time-name attribute specifies the name of the date-time field declaration (see section 9.11.4) that is used for all date-time fields (see section 9.10.3) that are displayed on the page.

<define name="common-presentation-header-footer-attlist" combine="interleave">

<optional>

<attribute name="presentation:use-date-time-name">

<ref name="string"/>

</attribute>

</optional>

</define>

#### ID

The draw:id attribute assigns a unique ID to a drawing page.

<define name="draw-page-attlist" combine="interleave">

<optional>

<attribute name="draw:id">

<ref name="ID"/>

</attribute>

</optional>

</define>

#### Navigation Order

The draw:nav-order attribute defines a logical navigation sequence for the graphical elements included in the page. Its value is a sequence of unique IDREFs. If this optional attribute is present, it must include all graphic elements not contained within a <draw:g> element. This attribute should reflect the intentional ordering of graphics as set by the document author.

<define name="draw-page-attlist" combine="interleave">

<optional>

<attribute name="draw:nav-order">

<ref name="IDREFS"/>

</attribute>

</optional>

</define>

### Presentation Notes

Each drawing page element in a presentation can have an additional presentation notes page, which contains a preview of the corresponding drawing page and additional graphic shapes. A notes page is described by the <presentation:notes> element, that may be contained in the <draw:page> element. See section 14.4.2 for more information about this element.

Example: Drawing page

<office:automatic-styles>

<style:style style:name="gg3434" style:family="drawing-page">

<style:drawing-page-properties presentation:page-duration="5s">

</style:style>

<style:style style:name="titledia"

style:family="presentation-page-layout">

<presentation:placeholder presentation:object="title"

svg:x="20%" svg:y="10%"

svg:width="80%" svg:height="10%"/>

<presentation:placeholder presentation:object="subtitle"

svg:x="20%" svg:y="30%"

svg:width="80%" svg:height="60%" />

</style:style>

</office:automatic-styles>

...

<office:body>

<draw:page office:name="Page 1" draw:style-name="gg3434"

draw:master-page-name="home"

presentation:page-layout-name="titledia">

<draw:rect .../>

presentation:notes>

<draw:text ...>this is a note</draw:text>

</presentation:notes>  
 </draw:page>

</office:body>

## Drawing Shapes

This section describes drawing shapes that might occur within all kind of applications.

<define name="shape">

<choice>

<ref name="draw-rect"/>

<ref name="draw-line"/>

<ref name="draw-polyline"/>

<ref name="draw-polygon"/>

<ref name="draw-regular-polygon"/>

<ref name="draw-path"/>

<ref name="draw-circle"/>

<ref name="draw-ellipse"/>

<ref name="draw-g"/>

<ref name="draw-page-thumbnail"/>

<ref name="draw-frame"/>

<ref name="draw-measure"/>

<ref name="draw-caption"/>

<ref name="draw-connector"/>

<ref name="draw-control"/>

<ref name="dr3d-scene"/>

<ref name="draw-custom-shape"/>

</choice>

</define>

### Rectangle

The <draw:rect> element represents a rectangular drawing shape.

<define name="draw-rect">

<element name="draw:rect">

<ref name="draw-rect-attlist"/>

<ref name="common-draw-position-attlist"/>

<ref name="common-draw-size-attlist"/>

<ref name="common-draw-shape-with-text-and-styles-attlist"/>

<ref name="common-draw-caption-id-attlist"/>

<optional>

<ref name="svg-title"/>

</optional>

<optional>

<ref name="svg-desc"/>

</optional>

<optional>

<ref name="office-event-listeners"/>

</optional>

<zeroOrMore>

<ref name="draw-glue-point"/>

</zeroOrMore>

<ref name="draw-text"/>

</element>

</define>

The attributes that may be associated with the <draw:rect> element are:

* Position, Size, Style, Layer, Z-Index, ID, Caption ID and Transformation – see section 9.2.15.
* Text anchor, table background, draw end position – see section 9.2.16.
* Round corners

The elements that may be contained in the <draw:rect> element are:

* Title (short accessible name) – see section 9.2.20.
* Long description (in support of accessibility) – see section 9.2.20.
* Event listeners – see section 9.2.21.
* Glue points – see section 9.2.19.
* Text – see section 9.2.17.

#### Round Corners

The attribute draw:corner-radius specifies the radius of the circle used to round off the corners of the rectangle.

<define name="draw-rect-attlist" combine="interleave">

<optional>

<attribute name="draw:corner-radius">

<ref name="nonNegativeLength"/>

</attribute>

</optional>

</define>

Example: Rectangular drawing shape

<draw:rect svg:x="2cm" svg:y="3cm" svg:width="10cm" svg:height="20cm" svg:transform="rotate(45)" draw:style-name="object-with-shadow">

### Line

The <draw:line> element represents a line.

<define name="draw-line">

<element name="draw:line">

<ref name="draw-line-attlist"/>

<ref name="common-draw-shape-with-text-and-styles-attlist"/>

<ref name="common-draw-caption-id-attlist"/>

<optional>

<ref name="svg-title"/>

</optional>

<optional>

<ref name="svg-desc"/>

</optional>

<optional>

<ref name="office-event-listeners"/>

</optional>

<zeroOrMore>

<ref name="draw-glue-point"/>

</zeroOrMore>

<ref name="draw-text"/>

</element>

</define>

The attributes that may be associated with the <draw:line> element are:

* Style, Layer, Z-Index, ID, Caption ID and Transformation – see section 9.2.15.
* Text anchor, table background, draw end position– see section 9.2.16.
* Start point
* End point

The elements that may be contained in the <draw:line> element are:

* Title (short accessible name) – see section 9.2.20.
* Long description (in support of accessibility) – see section 9.2.20.
* Event listeners – see section 9.2.21.
* Glue points – see section 9.2.19.
* Text – see section 9.2.17.

#### Start Point

The start point attributes svg:x1 and svg:y1 specify the start coordinates of the line.

<define name="draw-line-attlist" combine="interleave">

<attribute name="svg:x1">

<ref name="coordinate"/>

</attribute>

<attribute name="svg:y1">

<ref name="coordinate"/>

</attribute>

</define>

#### End Point

The end point attributes svg:x2 and svg:y2 specify the end coordinates of the line.

<define name="draw-line-attlist" combine="interleave">

<attribute name="svg:x2">

<ref name="coordinate"/>

</attribute>

<attribute name="svg:y2">

<ref name="coordinate"/>

</attribute>

</define>

### Polyline

The <draw:polyline> element represents a polyline drawing shape.

Some implementations may ignore the size attribute, and instead determine the size of a shape exclusively from the shape data (i.e., polygon vertices).

<define name="draw-polyline">

<element name="draw:polyline">

<ref name="common-draw-points-attlist"/>

<ref name="common-draw-position-attlist"/>

<ref name="common-draw-size-attlist"/>

<ref name="common-draw-viewbox-attlist"/>

<ref name="common-draw-shape-with-text-and-styles-attlist"/>

<ref name="common-draw-caption-id-attlist"/>

<optional>

<ref name="svg-title"/>

</optional>

<optional>

<ref name="svg-desc"/>

</optional>

<optional>

<ref name="office-event-listeners"/>

</optional>

<zeroOrMore>

<ref name="draw-glue-point"/>

</zeroOrMore>

<ref name="draw-text"/>

</element>

</define>

The attributes that may be associated with the <draw:polyline> element are:

* Position, Size, View box, Style, Layer, Z-Index, ID, Caption ID and Transformation – see section 9.2.15
* Text anchor, table background, draw end position – see section 9.2.16
* Points

The elements that may be contained in the <draw:polyline> element are:

* Title (short accessible name) – see section 9.2.20.
* Long description (in support of accessibility) – see section 9.2.20.
* Event listeners – see section 9.2.21.
* Glue points – see section 9.2.19.
* Text – see section 9.2.17.

#### Points

The draw:points attribute stores a sequence of points, which are connected by straight lines. Each point consists of two coordinates. The coordinates are separated by a comma and the points are separated by white spaces.

<define name="common-draw-points-attlist">

<attribute name="draw:points">

<ref name="points"/>

</attribute>

</define>

### Polygon

The <draw:polygon> element represents a polygon. A polygon is a closed set of straight lines.

Some implementations may ignore the size attribute, and instead determine the size of a shape exclusively from the shape data (i.e., polygon vertices).

<define name="draw-polygon">

<element name="draw:polygon">

<ref name="common-draw-points-attlist"/>

<ref name="common-draw-position-attlist"/>

<ref name="common-draw-size-attlist"/>

<ref name="common-draw-viewbox-attlist"/>

<ref name="common-draw-shape-with-text-and-styles-attlist"/>

<ref name="common-draw-caption-id-attlist"/>

<optional>

<ref name="svg-title"/>

</optional>

<optional>

<ref name="svg-desc"/>

</optional>

<optional>

<ref name="office-event-listeners"/>

</optional>

<zeroOrMore>

<ref name="draw-glue-point"/>

</zeroOrMore>

<ref name="draw-text"/>

</element>

</define>

The attributes that may be associated with the <draw:polygon> element are:

* Position, Size, View box, Style, Layer, Z-Index, ID, Caption ID and Transformation – see section 9.2.15
* Text anchor, table background, draw end position – see section 9.2.16
* Points – see section 9.2.3

The elements that may be contained in the <draw:polygon> element are:

* Title (short accessible name) – see section 9.2.20.
* Long description (in support of accessibility) – see section 9.2.20.
* Event listeners – see section 9.2.21.
* Glue points – see section 9.2.19.
* Text – see section 9.2.17.

### Regular Polygon

The <draw:regular-polygon> element represents a regular polygon. A regular polygon is a polygon that is specified by its number of edges (that is equal to the number of its corners), rather than by arbitrary points.

<define name="draw-regular-polygon">

<element name="draw:regular-polygon">

<ref name="draw-regular-polygon-attlist"/>

<ref name="common-draw-position-attlist"/>

<ref name="common-draw-size-attlist"/>

<ref name="common-draw-shape-with-text-and-styles-attlist"/>

<ref name="common-draw-caption-id-attlist"/>

<optional>

<ref name="svg-title"/>

</optional>

<optional>

<ref name="svg-desc"/>

</optional>

<optional>

<ref name="office-event-listeners"/>

</optional>

<zeroOrMore>

<ref name="draw-glue-point"/>

</zeroOrMore>

<ref name="draw-text"/>

</element>

</define>

The attributes that may be associated with the <draw:polygon> element are:

* Position, Size, Style, Layer, Z-Index, ID, Caption ID and Transformation – see section 9.2.15
* Text anchor, table background, draw end position – see section 9.2.16
* Concave
* Corners
* Sharpness

The elements that may be contained in the <draw:regular-polygon> element are:

* Title (short accessible name) – see section 9.2.20.
* Long description (in support of accessibility) – see section 9.2.20.
* Event listeners – see section 9.2.21.
* Glue points – see section 9.2.19.
* Text – see section 9.2.17.

#### Concave

The draw:concave attribute specifies whether the polygon is convex or concave. For a convex polygon, the polygon corners are located on a single ellipse which has its center in the center of the polygon. In a concave polygon, two such ellipses are required, and corners that are located next to each other are located on different ellipses. An example for a convex polygon is a hexagon. An example for a concave polygon is a star. For concave polygons, an additional draw:sharpness attribute is required.

<define name="draw-regular-polygon-attlist" combine="interleave">

<choice>

<attribute name="draw:concave">

<value>false</value>

</attribute>

<group>

<attribute name="draw:concave">

<value>true</value>

</attribute>

<ref name="draw-regular-polygon-sharpness-attlist"/>

</group>

</choice>

</define>

#### Corners

The draw:corners attribute specifies the number of polygon corners.

<define name="draw-regular-polygon-attlist" combine="interleave">

<attribute name="draw:corners">

<ref name="positiveInteger"/>

</attribute>

</define>

#### Sharpness

For concave attributes, the draw:sharpness attribute specifies the radius of the ellipse on which the inner polygon corners are located. The value is a percentage, where 0% means that all corners are located on a single ellipse, while 100% means that the inner corners are located at the center point of the polygon. In general, if *r* is the radius of the polygon, and *s* is the sharpness, the inner corners a located on a ellipse that's radius is *r(100-s)/100*.

<define name="draw-regular-polygon-sharpness-attlist">

<attribute name="draw:sharpness">

<ref name="percent"/>

</attribute>

</define>

### Path

The <draw:path> element represents a path. A path is a shape with a user-defined outline. The shape is built using multiple drawing actions such as:

* moveto – set a new current point
* lineto – draw a straight line
* curveto – draw a curve using a cubic Bézier
* arc – draw an elliptical or circular arc
* closepath – close the current shape by drawing a line to the last moveto

Compound paths are paths with subpaths, each subpath consisting of a single moveto followed by one or more line or curve operations. Compound paths can be used for effects such as holes in objects.

Some implementations may ignore the size attribute, and instead determine the size of a shape exclusively from the shape data (i.e., polygon vertices).

<define name="draw-path">

<element name="draw:path">

<ref name="common-draw-path-data-attlist"/>

<ref name="common-draw-position-attlist"/>

<ref name="common-draw-size-attlist"/>

<ref name="common-draw-viewbox-attlist"/>

<ref name="common-draw-shape-with-text-and-styles-attlist"/>

<ref name="common-draw-caption-id-attlist"/>

<optional>

<ref name="svg-title"/>

</optional>

<optional>

<ref name="svg-desc"/>

</optional>

<optional>

<ref name="office-event-listeners"/>

</optional>

<zeroOrMore>

<ref name="draw-glue-point"/>

</zeroOrMore>

<ref name="draw-text"/>

</element>

</define>

The attributes that may be associated with the <draw:path> element are:

* Position, Size, View box, Style, Layer, Z-Index, ID, Caption ID and Transformation – see section 9.2.15
* Text anchor, table background, draw end position – see section 9.2.16
* Path data

The elements that may be contained in the <draw:path> element are:

* Title (short accessible name) – see section 9.2.20.
* Long description (in support of accessibility) – see section 9.2.20.
* Event listeners – see section 9.2.21.
* Glue points – see section 9.2.19.
* Text – see section 9.2.17.

#### Path Data

The syntax for the attribute svg:d is described in §8 of the *Scalable Vector Graphics (SVG) 1.1 Specification* [SVG].

Some implementations may only supports a subset of the SVG path specification, for instance no mixtures of open and closed curves for one shape, or no elliptical arc command.

<define name="common-draw-path-data-attlist">

<attribute name="svg:d">

<ref name="pathData"/>

</attribute>

</define>

### Circle

The <draw:circle> element represents a circular drawing shape.

<define name="draw-circle">

<element name="draw:circle">

<ref name="draw-circle-attlist"/>

<ref name="common-draw-circle-ellipse-attlist"/>

<ref name="common-draw-position-attlist"/>

<ref name="common-draw-size-attlist"/>

<ref name="common-draw-shape-with-text-and-styles-attlist"/>

<ref name="common-draw-caption-id-attlist"/>

<optional>

<ref name="svg-title"/>

</optional>

<optional>

<ref name="svg-desc"/>

</optional>

<optional>

<ref name="office-event-listeners"/>

</optional>

<zeroOrMore>

<ref name="draw-glue-point"/>

</zeroOrMore>

<ref name="draw-text"/>

</element>

</define>

The attributes that may be associated with the <draw:circle> element are:

* Position, Size, Style, Layer, Z-Index, ID, Caption ID and Transformation – see section 9.2.15
* Text anchor, table background, draw end position – see section 9.2.16
* Center point
* Radius
* Kind
* Start angle
* End angle

The elements that may be contained in the <draw:circle> element are:

* Title (short accessible name) – see section 9.2.20.
* Long description (in support of accessibility) – see section 9.2.20.
* Event listeners – see section 9.2.21.
* Glue points – see section 9.2.19.
* Text – see section 9.2.17.

#### Center Point

The center point attributes svg:cx and svg:cy specify the coordinates of the center point of the circle. If these optional attributes are not set, the position and size attributes are used to create them.

<define name="common-draw-circle-ellipse-attlist" combine="interleave">

<optional>

<attribute name="svg:cx">

<ref name="coordinate"/>

</attribute>

<attribute name="svg:cy">

<ref name="coordinate"/>

</attribute>

</optional>

</define>

#### Radius

The svg:r attribute specifies the radius of the circle. If this optional attribute are not set, the position and size attributes are used to create circle.

<define name="draw-circle-attlist" combine="interleave">

<optional>

<attribute name="svg:r">

<ref name="length"/>

</attribute>

</optional>

</define>

#### Kind

The draw:kind attribute specifies the appearance of the circle.

* full specifies a full circle or ellipse, like .
* section specifies a section of a circle or ellipse, like .
* cut specifies a circle or ellipse with a cut, like .
* arc specifies a circle or ellipse arc, like .

<define name="common-draw-circle-ellipse-attlist" combine="interleave">

<optional>

<attribute name="draw:kind" a:defaultValue="full">

<choice>

<value>full</value>

<value>section</value>

<value>cut</value>

<value>arc</value>

</choice>

</attribute>

</optional>

</define>

#### Start Angle

For circles where the draw:kind attribute value is section, cut or arc, the svg:start-angle attribute specifies the start angle of the section, cut, or arc.

<define name="common-draw-circle-ellipse-attlist" combine="interleave">

<optional>

<attribute name="draw:start-angle">

<ref name="double"/>

</attribute>

</optional>

</define>

#### End Angle

For circles where the draw:kind attribute value is section, cut or arc, the svg:end-angle attribute specifies the end angle of the section, cut, or arc.

<define name="common-draw-circle-ellipse-attlist" combine="interleave">

<optional>

<attribute name="draw:end-angle">

<ref name="double"/>

</attribute>

</optional>

</define>

### Ellipse

The <draw:ellipse> element represents an ellipse.

<define name="draw-ellipse">

<element name="draw:ellipse">

<ref name="common-draw-circle-ellipse-attlist"/>

<ref name="draw-ellipse-attlist"/>

<ref name="common-draw-position-attlist"/>

<ref name="common-draw-size-attlist"/>

<ref name="common-draw-shape-with-text-and-styles-attlist"/>

<ref name="common-draw-caption-id-attlist"/>

<optional>

<ref name="svg-title"/>

</optional>

<optional>

<ref name="svg-desc"/>

</optional>

<optional>

<ref name="office-event-listeners"/>

</optional>

<zeroOrMore>

<ref name="draw-glue-point"/>

</zeroOrMore>

<ref name="draw-text"/>

</element>

</define>

The attributes that may be associated with the <draw:ellipse> element are:

* Position, Size, Style, Layer, Z-Index, ID, Caption ID and Transformation – see section 9.2.15
* Text anchor, table background, draw end position – see section 9.2.16
* Center point, Kind, Start angle, End angle – see section 9.2.7
* Radius

The elements that may be contained in the <draw:ellipse> element are:

* Title (short accessible name) – see section 9.2.20.
* Long description (in support of accessibility) – see section 9.2.20.
* Event listeners – see section 9.2.21.
* Glue points – see section 9.2.19.
* Text – see section 9.2.17.

#### Radius

The svg:rx and svg:rx attribute specify the horizontal and vertical radius of the ellipse. If these optional attributes are not set, the position and size attributes are used to create the ellipse.

<define name="draw-ellipse-attlist" combine="interleave">

<optional>

<attribute name="svg:rx">

<ref name="length"/>

</attribute>

<attribute name="svg:ry">

<ref name="length"/>

</attribute>

</optional>

</define>

### Connector

The <draw:connector> element represents a series of lines that are connected to the glue points of two other shapes.

<define name="draw-connector">

<element name="draw:connector">

<ref name="draw-connector-attlist"/>

<ref name="common-draw-shape-with-text-and-styles-attlist"/>

<ref name="common-draw-caption-id-attlist"/>

<optional>

<ref name="svg-title"/>

</optional>

<optional>

<ref name="svg-desc"/>

</optional>

<optional>

<ref name="office-event-listeners"/>

</optional>

<zeroOrMore>

<ref name="draw-glue-point"/>

</zeroOrMore>

<ref name="draw-text"/>

</element>

</define>

The attributes that may be associated with the <draw:connector> element are:

* Style, Layer, Z-Index, ID and Caption ID – see section 9.2.15
* Text anchor, table background, draw end position – see section 9.2.16
* Type
* Start position
* Start shape
* Start glue point
* End position
* End shape
* End glue point
* Line skew

The elements that may be contained in the <draw:connector> element are:

* Title (short accessible name) – see section 9.2.20.
* Long description (in support of accessibility) – see section 9.2.20.
* Event listeners – see section 9.2.21.
* Glue points – see section 9.2.19.
* Text – see section 9.2.17.

#### Type

The draw:type attribute specifies how the connection between two points is rendered. The value of this attribute can be standard, lines, line, or curve.

* standard: a standard connector escapes the two connecting objects with straight lines and connects them with a straight perpendicular line.
* lines: a lines connector escapes the two connecting objects with straight lines and connects them with a straight (not necessarily perpendicular) line.
* line: a line connector draws one straight line between the two escape points of the connected objects.
* curve: a curve connector draws a single curved line between the two escape points of the connected objects.

<define name="draw-connector-attlist" combine="interleave">

<optional>

<attribute name="draw:type" a:defaultValue="standard">

<choice>

<value>standard</value>

<value>lines</value>

<value>line</value>

<value>curve</value>

</choice>

</attribute>

</optional>

</define>

#### Start Position

The start position attributes svg:x1 and svg:y1 specify the start position of a connector.

If the start position is connected to a shape, these attributes are optional because the start position defaults to the corresponding glue point on the target shape.

<define name="draw-connector-attlist" combine="interleave">

<optional>

<attribute name="svg:x1">

<ref name="coordinate"/>

</attribute>

<attribute name="svg:y1">

<ref name="coordinate"/>

</attribute>

</optional>

</define>

#### Start Shape

The draw:start-shape attribute identifies the drawing shape to which the start of this connector is connected by its name.

If a shape is connected to the start of a connector, the start position defaults to the corresponding glue point on the target shape.

<define name="draw-connector-attlist" combine="interleave">

<optional>

<attribute name="draw:start-shape">

<ref name="IDREF"/>

</attribute>

</optional>

</define>

#### Start Glue Point

The draw:start-glue-point attribute identifies the glue point in the start shape of the connector by its number. See section 9.2.19 for details on glue points.

If this attribute is not set and the start of the connector is connected to a shape, the application may choose the glue point. If the start of the connector is not connected to a shape, this attribute is ignored.

<define name="draw-connector-attlist" combine="interleave">

<optional>

<attribute name="draw:start-glue-point">

<ref name="nonNegativeInteger"/>

</attribute>

</optional>

</define>

#### End Position

The end position attributes svg:x2 and svg:y2 specify the end position of a connector.

If the end position is connected to a shape, these attributes are optional because the end position defaults to the corresponding glue point on the target shape.

<define name="draw-connector-attlist" combine="interleave">

<optional>

<attribute name="svg:x2">

<ref name="coordinate"/>

</attribute>

<attribute name="svg:y2">

<ref name="coordinate"/>

</attribute>

</optional>

</define>

#### End Shape

The draw:end-shape attribute identifies the drawing shape to which the end of the connector is connected by its name.

If a shape is connected to the end of a connector, the end position defaults to the corresponding glue point on the target shape.

<define name="draw-connector-attlist" combine="interleave">

<optional>

<attribute name="draw:end-shape">

<ref name="IDREF"/>

</attribute>

</optional>

</define>

#### End Glue Point

The draw:end-glue-point attribute identifies the glue point in the end shape of the connector by its number. See section 9.2.19 for details on glue points.

If this attribute is not set and the end of the connector is connected to a shape, the application may choose the glue point. If the end of the connector is not connected to a shape, this attribute is ignored.

<define name="draw-connector-attlist" combine="interleave">

<optional>

<attribute name="draw:end-glue-point">

<ref name="nonNegativeInteger"/>

</attribute>

</optional>

</define>

#### Line Skew

The draw:line-skew attribute controls the generation of the lines that connect the start and end points. Depending on the type of connector, this can vary from one to three distances that move the connector lines relative to their normal position.

<define name="draw-connector-attlist" combine="interleave">

<optional>

<attribute name="draw:line-skew">

<list>

<ref name="length"/>

<optional>

<ref name="length"/>

<optional>

<ref name="length"/>

</optional>

</optional>

</list>

</attribute>

</optional>

</define>

### Caption

The <draw:caption> element represents a rectangular drawing shape with an additional set of lines. It can be used as a description for a fixed point inside a drawing.

<define name="draw-caption">

<element name="draw:caption">

<ref name="draw-caption-attlist"/>

<ref name="common-draw-position-attlist"/>

<ref name="common-draw-size-attlist"/>

<ref name="common-draw-shape-with-text-and-styles-attlist"/>

<ref name="common-draw-caption-id-attlist"/>

<optional>

<ref name="svg-title"/>

</optional>

<optional>

<ref name="svg-desc"/>

</optional>

<optional>

<ref name="office-event-listeners"/>

</optional>

<zeroOrMore>

<ref name="draw-glue-point"/>

</zeroOrMore>

<ref name="draw-text"/>

</element>

</define>

The attributes that may be associated with the <draw:caption> element are:

* Position, Size, Style, Layer, Z-Index, ID, Caption ID and Transformation – see section 9.2.15
* Text anchor, table background, draw end position – see section 9.2.16
* Caption point
* Round corners

The elements that may be contained in the <draw:caption> element are:

* Title (short accessible name) – see section 9.2.20.
* Long description (in support of accessibility) – see section 9.2.20.
* Event listeners – see section 9.2.21.
* Glue points – see section 9.2.19.
* Text – see section 9.2.17.

#### Caption Point

The caption point attributes draw:caption-point-x and draw:caption-point-y specify the position of the point that is captioned. A set of lines are rendered from the caption area.

<define name="draw-caption-attlist" combine="interleave">

<optional>

<attribute name="draw:caption-point-x">

<ref name="coordinate"/>

</attribute>

<attribute name="draw:caption-point-y">

<ref name="coordinate"/>

</attribute>

</optional>

</define>

#### Round Corners

The draw:corner-radius attribute specifies the radius of the circle used to round off the corners of the caption.

<define name="draw-caption-attlist" combine="interleave">

<optional>

<attribute name="draw:corner-radius">

<ref name="nonNegativeLength"/>

</attribute>

</optional>

</define>

### Measure

The <draw:measure> element represents a shape that is used to measure distances in drawings.

<define name="draw-measure">

<element name="draw:measure">

<ref name="draw-measure-attlist"/>

<ref name="common-draw-shape-with-text-and-styles-attlist"/>

<ref name="common-draw-caption-id-attlist"/>

<optional>

<ref name="svg-title"/>

</optional>

<optional>

<ref name="svg-desc"/>

</optional>

<optional>

<ref name="office-event-listeners"/>

</optional>

<zeroOrMore>

<ref name="draw-glue-point"/>

</zeroOrMore>

<ref name="draw-text"/>

</element>

</define>

The attributes that may be associated with the <draw:measure> element are:

* Style, Layer, Z-Index, ID, Caption ID and Transformation – see section 9.2.15
* Text anchor, table background, draw end position – see section 9.2.16
* Start position
* End position

The elements that may be contained in the <draw:measure> element are:

* Title (short accessible name) – see section 9.2.20.
* Long description (in support of accessibility) – see section 9.2.20.
* Event listeners – see section 9.2.21.
* Glue points – see section 9.2.19.
* Text – see section 9.2.17.

#### Start Position

The attributes svg:x1 and svg:y1 specify the start point of the measured distance.

<define name="draw-measure-attlist" combine="interleave">

<attribute name="svg:x1">

<ref name="coordinate"/>

</attribute>

<attribute name="svg:y1">

<ref name="coordinate"/>

</attribute>

</define>

#### Draw End Position

The attributes svg:x2 and svg:y2 specify the end point of the measured distance.

<define name="draw-measure-attlist" combine="interleave">

<attribute name="svg:x2">

<ref name="coordinate"/>

</attribute>

<attribute name="svg:y2">

<ref name="coordinate"/>

</attribute>

</define>

### Control

The <draw:control> element represents a shape that is linked to a control inside an <office:forms> element (see section 11.1).

<define name="draw-control">

<element name="draw:control">

<ref name="draw-control-attlist"/>

<ref name="common-draw-position-attlist"/>

<ref name="common-draw-size-attlist"/>

<ref name="common-draw-shape-with-text-and-styles-attlist"/>

<ref name="common-draw-caption-id-attlist"/>

<optional>

<ref name="svg-title"/>

</optional>

<optional>

<ref name="svg-desc"/>

</optional>

<zeroOrMore>

<ref name="draw-glue-point"/>

</zeroOrMore>

</element>

</define>

The attributes that may be associated with the <draw:control> element are:

* Position, Size, Style, Layer, Z-Index, ID, Caption ID and Transformation – see section 9.2.15.
* Text anchor, table background, draw end position – see section 9.2.16
* Control

The elements that may be contained in the <draw:control> element are:

* Title (short accessible name) – see section 9.2.20.
* Long description (in support of accessibility) – see section 9.2.20.
* Glue points – see section 9.2.19.

#### Control

The attributes draw:control attribute specifies the control within a form (see section 11.5.2) that is linked to the control shape.

<define name="draw-control-attlist" combine="interleave">

<attribute name="draw:control">

<ref name="IDREF"/>

</attribute>

</define>

### Page Thumbnail

The <draw:page-thumbnail> element represents a rectangular area that displays the thumbnail of a drawing page.

<define name="draw-page-thumbnail">

<element name="draw:page-thumbnail">

<ref name="draw-page-thumbnail-attlist"/>

<ref name="common-draw-position-attlist"/>

<ref name="common-draw-size-attlist"/>

<ref name="presentation-shape-attlist"/>

<ref name="common-draw-shape-with-styles-attlist"/>

<ref name="common-draw-caption-id-attlist"/>

<optional>

<ref name="svg-title"/>

</optional>

<optional>

<ref name="svg-desc"/>

</optional>

</element>

</define>

The attributes that may be associated with the <draw:page-thumbnail> element are:

* Position, Size, Style, Layer, Z-Index, ID, Caption ID and Transformation – see section 9.2.15.
* Text anchor, table background, draw end position – see section 9.2.16
* Presentation class – see section 9.6.1
* Page number

The elements that may be contained in the <draw:page-thumbnail> element are:

* Title (short accessible name) – see section 9.2.20.
* Long description (in support of accessibility) – see section 9.2.20.

#### Page Number

The draw:page-number attribute specifies the number of the page that is displayed as a thumbnail. For thumbnails on notes pages, the value of this attribute is fixed to the drawing page of the notes page. For thumbnails on handout master pages, the value of this attribute is the order in which the pages are previewed on the handout. For example, on a handout page with 4 thumbnails, the thumbnail with the lowest page number displays the first page when printing the first handout page and the fifth page when printing the second handout page and so on.

<define name="draw-page-thumbnail-attlist">

<optional>

<attribute name="draw:page-number">

<ref name="positiveInteger"/>

</attribute>

</optional>

</define>

### Grouping

The <draw:g> element represents a group of drawing shapes.

<define name="draw-g">

<element name="draw:g">

<ref name="draw-g-attlist"/>

<ref name="common-draw-z-index-attlist"/>

<ref name="common-draw-name-attlist"/>

<ref name="common-draw-id-attlist"/>

<ref name="common-draw-style-name-attlist"/>

<ref name="common-text-spreadsheet-shape-attlist"/>

<ref name="common-draw-caption-id-attlist"/>

<optional>

<ref name="svg-title"/>

</optional>

<optional>

<ref name="svg-desc"/>

</optional>

<optional>

<ref name="office-event-listeners"/>

</optional>

<zeroOrMore>

<ref name="draw-glue-point"/>

</zeroOrMore>

<zeroOrMore>

<ref name="shape"/>

</zeroOrMore>

</element>

</define>

The attributes that may be associated with the <draw:g> element are:

* Style, Z-Index, ID and Caption ID – see section 9.2.15.
* Text anchor, table background, draw end position – see section 9.2.16
* Position

The elements that may be contained in the <draw:g> element are:

* Title (short accessible name) – see section 9.2.20.
* Long description (in support of accessibility) – see section 9.2.20.
* Event listeners – see section 9.2.21.
* Glue points – see section 9.2.19.
* Drawing shapes.

#### Position

For group shapes that are contained in text documents and anchored as character, the svg:y attribute specifies the vertical position of the shape.

<define name="draw-g-attlist" combine="interleave">

<optional>

<attribute name="svg:y">

<ref name="coordinate"/>

</attribute>

</optional>

</define>

### Common Drawing Shape Attributes

The attributes described in this section are common to all drawing shapes.

#### Name

The attribute draw:name assigns a name to the drawing shape.

<define name="common-draw-name-attlist" combine="interleave">

<optional>

<attribute name="draw:name">

<ref name="string"/>

</attribute>

</optional>

</define>

#### Caption-ID

The draw:caption-id attribute establishes a relationship between a drawing objects and its caption. It takes a value of type IDREF. The value for draw:caption-id attribute is the target ID assigned to the <draw:text-box> (see section 9.3.1) used to represent the corresponding caption.

When a caption is assigned by a user agent, an id must be assigned to the element containing the text used to caption a drawing element. The drawing element being captioned must then be assigned the draw:caption-id attribute with an IDREF equivalent to the id <draw:text-box> containing the captioning text, thus establishing a relationship between the captioned text and the object captioned as needed for accessibility. Removing the caption should result in removing the draw:caption-id attribute of the object that was being captioned.

If the user agent supports a platform which provides a draw:caption-id relationship in its accessibility API, this relationship for captions should be used to fulfill the relationship.

See appendix E for guidelines how to use this attribute.

<define name="common-draw-caption-id-attlist" combine="interleave">

<optional>

<attribute name="draw:caption-id">

<ref name="IDREF"/>

</attribute>

</optional>

</define>

#### Position

The position attributes svg:x and svg:y specify the x and y coordinates of the start position of the drawing shape.

<define name="common-draw-position-attlist">

<optional>

<attribute name="svg:x">

<ref name="coordinate"/>

</attribute>

</optional>

<optional>

<attribute name="svg:y">

<ref name="coordinate"/>

</attribute>

</optional>

</define>

#### Size

The attributes svg:width and svg:height specify the width and height of the drawing shape.

<define name="common-draw-size-attlist">

<optional>

<attribute name="svg:width">

<ref name="length"/>

</attribute>

</optional>

<optional>

<attribute name="svg:height">

<ref name="length"/>

</attribute>

</optional>

</define>

#### Transformation

The draw:transform attribute specifies a list of transformations that can be applied to a drawing shape.

The value of this attribute is a list of transform definitions, which are applied to the drawing shape in the order in which they are listed. The transform definitions in the list must be separated by a white space and/or a comma. The types of transform definitions available include:

* matrix(<a> <b> <c> <d> <e> <f>), which specifies a transformation in the form of a transformation matrix of six values. matrix(a,b,c,d,e,f) is the equivalent of applying the transformation matrix [a b c d e f].
* translate(<tx> [<ty>]), which specifies a translation by tx and ty.
* scale(<sx> [<sy>]), which specifies a scale operation by sx and sy. If <sy> is not provided, it is assumed to be equal to <sx>.
* rotate(<rotate-angle>), which specifies a rotation by <rotate-angle> about the origin of the shapes coordinate system.
* skewX(<skew-angle>), which specifies a skew transformation along the X axis.
* skewY(<skew-angle>), which specifies a skew transformation along the Y axis.

<define name="common-draw-transform-attlist">

<optional>

<attribute name="draw:transform">

<ref name="string"/>

</attribute>

</optional>

</define>

#### View Box

The svg:viewBox attribute establishes a user coordinate system inside the physical coordinate system of the shape specified by the position and size attributes. This user coordinate system is used by the svg:points attribute and the <draw:path> element.

The syntax for using this attribute is the same as the [SVG] syntax. The value of the attribute are four numbers separated by white spaces, which define the left, top, right, and bottom dimensions of the user coordinate system.

Some implementations may ignore the view box attribute. The implied coordinate system then has its origin at the left, top corner of the shape, without any scaling relative to the shape.

<define name="common-draw-viewbox-attlist">

<attribute name="svg:viewBox">

<list>

<ref name="integer"/>

<ref name="integer"/>

<ref name="integer"/>

<ref name="integer"/>

</list>

</attribute>

</define>

#### Style

The draw:style-name and presentation:style-name attributes specify a style for the drawing shape. If draw:style-name is used, the shape is a regular graphic shape. If presentation:style-name is used, the shape is a presentation shape as described in section 9.6.

The value of both attributes is the name of a <style:style> element. If the draw:style-name attribute is used, the style must have a family value of graphic. If the presentation:style-name is used, the style must have a family value of presentation. The formatting properties of the specified style and its optional parent styles are used to format the shape. See also section 14.13.1.

The draw:class-names and presentation:class-names attributes take a whitespace separated list of either graphic or presentation style names. The referenced styles are applied in the order they are contained in the list. If both, draw:style-name and draw:class-names, or both presentation:style-name and presentation:class-names are present, the style referenced by the style-name attribute is treated as the first style in the list in the class-names attribute. Conforming application should support the class-names attribute and also should preserve it while editing.

<define name="common-draw-style-name-attlist">

<choice>

<group>

<optional>

<attribute name="draw:style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

<optional>

<attribute name="draw:class-names">

<ref name="styleNameRefs"/>

</attribute>

</optional>

</group>

<group>

<optional>

<attribute name="presentation:style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

<optional>

<attribute name="presentation:class-names">

<ref name="styleNameRefs"/>

</attribute>

</optional>

</group>

</choice>

</define>

#### Text Style

The draw:text-style-name attribute specifies a style for the drawing shape that is used to format the text that can be added to this shape.

The value of this attribute is the name of a <style:style> element with a family value of paragraph.

<define name="common-draw-text-style-name-attlist">

<optional>

<attribute name="draw:text-style-name">

<ref name="styleNameRef"/>

</attribute>

</optional>

</define>

#### Layer

The attribute draw:layer can assign each shape to a layer. The value of this attribute must be the name of a layer inside the layer-set of the document.

<define name="common-draw-layer-name-attlist">

<optional>

<attribute name="draw:layer">

<data type="string"/>

</attribute>

</optional>

</define>

#### ID

The draw:id attribute assigns an unique ID to a drawing shape that can be used to reference the shape.

<define name="common-draw-id-attlist">

<optional>

<attribute name="draw:id">

<ref name="ID"/>

</attribute>

</optional>

</define>

#### Z-Index

Drawing shapes are rendered in a specific order. In general, the shapes are rendered in the order in which they appear in the XML document. To change the order, use the svg:z-index attribute.

This attribute is optional.

<define name="common-draw-z-index-attlist">

<optional>

<attribute name="draw:z-index">

<ref name="nonNegativeInteger"/>

</attribute>

</optional>

</define>

### Common Shape Attributes for Text and Spreadsheet Documents

The attributes described in this section are common to all drawing shapes contained in text and spreadsheet documents.

#### End Position

If a drawing shape is included in a spreadsheet document and if the anchor of the shape is in a cell, then the attributes table:end-cell-address, table:end-x and table:end-y specify the end position of the shape and the size attributes are ignored. The end position is specified using the cell address of the cell in which the end position is located, and the x and y coordinates of the end position relative to the top left edge of the cell.

<define name="common-text-spreadsheet-shape-attlist" combine="interleave">

<optional>

<attribute name="table:end-cell-address">

<ref name="cellAddress"/>

</attribute>

</optional>

<optional>

<attribute name="table:end-x">

<ref name="coordinate"/>

</attribute>

</optional>

<optional>

<attribute name="table:end-y">

<ref name="coordinate"/>

</attribute>

</optional>

</define>

#### Table Background

If a drawing shape is included in a spreadsheet document, then the table:table-background attribute specifies whether or not the shape is in the table background. If the attribute is not existing, the shape is included in the foreground of the table.

<define name="common-text-spreadsheet-shape-attlist" combine="interleave">

<optional>

<attribute name="table:table-background">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Text Anchor

Within text documents, the anchor type attribute text:anchor-type specifies how a frame is bound to the text document. The anchor position is the point at which a frame is bound to a text document. The anchor position depends on the anchor type as explained in the following table.

| If the value of the text:anchor-type attribute is ... | The anchor position is... | The drawing shape element appears ... | Notes |
| --- | --- | --- | --- |
| page | The page that has the same physical page number as the value of the text:anchor-page-number attribute that is attached to the drawing shape element. If no text:anchor-page-number attribute is given, the anchor position is the page at which the character behind the drawing object element appears. | Either   * At the start of the document body, outside any paragraph or frame, provided a text:anchor-page-number attribute is given.   Or   * Inside any paragraph element that is not contained in a header, footer, footnote, or text box, if a text:anchor-page-number attribute is not given. | The physical page number is the number assigned to the page if all pages in the document are counted starting with page 1. |
| frame | The parent text box that the current drawing shape element is contained in. | In the element representing the text box to which the drawing object is bound. For example, if an image is bound to a text box, the image element is located in the text box element. |  |
| paragraph | The paragraph that the current drawing shape element is contained in. | At the start of the paragraph element. |  |
| char | The character after the drawing shape element. | Just before the character. |  |
| as-char | There is no anchor position. The drawing shape behaves like a character. | At the position where the character appears in the document. |  |

<define name="common-text-spreadsheet-shape-attlist" combine="interleave">

<ref name="common-text-anchor-attlist"/>

</define>

<define name="common-text-anchor-attlist" combine="interleave">

<optional>

<attribute name="text:anchor-type">

<choice>

<value>page</value>

<value>frame</value>

<value>paragraph</value>

<value>char</value>

<value>as-char</value>

</choice>

</attribute>

</optional>

</define>

#### Anchor Page Number

Within text documents, the text:anchor-page-number attribute specifies the physical page number of an anchor if the drawing object is bound to a page.

<define name="common-text-anchor-attlist" combine="interleave">

<optional>

<attribute name="text:anchor-page-number">

<ref name="positiveInteger"/>

</attribute>

</optional>

</define>

### Common Drawing Shape Content

Most drawing shapes may contain text content. The text content may contain paragraphs (see section 4.1.2) as well as lists (see section 4.3).

<define name="draw-text">

<zeroOrMore>

<choice>

<ref name="text-p"/>

<ref name="text-list"/>

</choice>

</zeroOrMore>

</define>

### Common Shape Attribute Groups

The following defined attributes are common for all shapes that supports styles and no text.

<define name="common-draw-shape-with-styles-attlist">

<ref name="common-draw-z-index-attlist"/>

<ref name="common-draw-id-attlist"/>

<ref name="common-draw-layer-name-attlist"/>

<ref name="common-draw-style-name-attlist"/>

<ref name="common-draw-transform-attlist"/>

<ref name="common-draw-name-attlist"/>

<ref name="common-text-spreadsheet-shape-attlist"/>

</define>

The following defined attributes are common for all shapes that supports styles and text.

<define name="common-draw-shape-with-text-and-styles-attlist">

<ref name="common-draw-shape-with-styles-attlist"/>

<ref name="common-draw-text-style-name-attlist"/>

</define>

### Glue Points

Glue points are designated points on the area of a drawing object to which a connector shape can connect. Most drawing objects have four standard glue points at the four edges of the object. Additional glue points may be added to a drawing object by inserting one or more <draw:glue-point> elements into a drawing object element. A <draw:glue-point> element creates a single user-defined glue point if placed inside a drawing object element, for example, a <draw:rectangle> element.

<define name="draw-glue-point">

<element name="draw:glue-point">

<ref name="draw-glue-point-attlist"/>

<empty/>

</element>

</define>

#### ID

The draw:id attribute contains the id of the glue point. The id a number and is used inside the draw:start-glue-point and draw:end-glue-point attributes of a <draw:connector> element. The Ids 0 to 3 are reserved for the 4 standard glue points that most drawing objects have. The glue points are numbered clockwise, starting at the top left corner of the shape.

<define name="draw-glue-point-attlist" combine="interleave">

<attribute name="draw:id">

<ref name="nonNegativeInteger"/>

</attribute>

</define>

#### Position

The svg:x and svg:y attributes specifies the position of the glue point. The coordinates are either percentage values relative to the drawing objects center or, if the draw:align attribute is also specified, absolute distance values relative to the edge specified with the draw:align attribute.

<define name="draw-glue-point-attlist" combine="interleave">

<attribute name="svg:x">

<choice>

<ref name="distance"/>

<ref name="percent"/>

</choice>

</attribute>

<attribute name="svg:y">

<choice>

<ref name="distance"/>

<ref name="percent"/>

</choice>

</attribute>

</define>

#### Align

The attribute draw:align specifies the alignment behavior of the glue point if the drawing object is resized and the shape edge to which the glue point's position relates. A missing vertical or horizontal position in the attribute's value means that the glue point is horizontally or vertically centered.

<define name="draw-glue-point-attlist" combine="interleave">

<optional>

<attribute name="draw:align">

<choice>

<value>top-left</value>

<value>top</value>

<value>top-right</value>

<value>left</value>

<value>center</value>

<value>right</value>

<value>bottom-left</value>

<value>bottom-right</value>

</choice>

</attribute>

</optional>

</define>

#### Escape Direction

The attribute draw:escape-direction specifies the direction in which the connection line escapes from the drawing object if a connector connects to the glue point. The value horizontal means the the connection line may escape to the left or to the right, the value vertical means that the connection line may escape up or down. The value auto means that the connection line may escape in all four directions.

<define name="draw-glue-points-attlist" combine="interleave">

<attribute name="draw:escape-direction">

<choice>

<value>auto</value>

<value>left</value>

<value>right</value>

<value>up</value>

<value>down</value>

<value>horizontal</value>

<value>vertical</value>

</choice>

</attribute>

</define>

### Title and Description

The <svg:title> and <svg:desc> elements specify text-only description strings for graphical objects as specified in §5.4 of [SVG].

The <svg:title> element is used as a short accessible name.

<define name="svg-title">

<element name="svg:title">

<text/>

</element>

</define>

The <svg:desc> element is used for the long description in support of accessibility.

<define name="svg-desc">

<element name="svg:desc">

<text/>

</element>

</define>

See appendix E for guidelines how to use these elements.

The <svg:title> and <svg:desc> elements can be used with the following drawing shape elements:

* <draw:rect>
* <draw:line>
* <draw:polyline>
* <draw:polygon>
* <draw:regular-polygon>
* <draw:path>
* <draw:circle>
* <draw:ellipse>
* <draw:g>
* <draw:page-thumbnail>
* <draw:frame>
* <draw:measure>
* <draw:caption>
* <draw:connector>
* <draw:control>
* <dr3d:scene>
* <draw:custom-shape>

It is further supported by layers (see section 9.1.3) and client side image maps (see section 9.3.10).

### Event Listeners

Drawing shapes may have event listeners attached. The event listeners that are attached to, for example, a text box or an image, are represented by an event listener element as described in section 12.4. This element is contained within the drawing object element, for example, the <draw:text-box> element or the <draw:image> element.

## Frames

A **frame** is a rectangular container where that contains enhanced content like text boxes, images or objects. Frames are very similar to regular drawing shapes, but support some features that are not available for regular drawing shapes, like contours, image maps and hyperlinks. In particular, a frame allows to have multiple renditions of an object. That is, a frame may for instance contain an object as well as an image. In this case, the application may choose the content that it supports best. If the application supports the object type contained in the frame, it probably will render the object. If it does not support the object, it will render the image.

In general, an application must not render more than one of the content elements contained in a frame. The order of content elements dictates the document author's preference for rendering, with the first child being the most preferred. This means that applications should render the first child element that it supports. A frame must contain at least one content element. The inclusion of multiple content elements is optional. Application may preserve the content elements they don't render, but don't have to.

Within text documents, frames are also used to position content outside the default text flow of a document.

Frames can contain:

* Text boxes
* Objects represented either in the OpenDocument format or in a object specific binary format
* Images
* Applets
* Plug-ins
* Floating frames

Like the formatting properties of drawing shapes, frame formatting properties are stored in styles belonging to the graphic family. The way a frame is contained in a document also is the same as for drawing shapes.

<define name="draw-frame">

<element name="draw:frame">

<ref name="common-draw-shape-with-text-and-styles-attlist"/>

<ref name="common-draw-position-attlist"/>

<ref name="common-draw-rel-size-attlist"/>

<ref name="common-draw-caption-id-attlist"/>

<ref name="presentation-shape-attlist"/>

<ref name="draw-frame-attlist"/>

<zeroOrMore>

<choice>

<ref name="draw-text-box"/>

<ref name="draw-image"/>

<ref name="draw-object"/>

<ref name="draw-object-ole"/>

<ref name="draw-applet"/>

<ref name="draw-floating-frame"/>

<ref name="draw-plugin"/>

</choice>

</zeroOrMore>

<optional>

<ref name="office-event-listeners"/>

</optional>

<zeroOrMore>

<ref name="draw-glue-point"/>

</zeroOrMore>

<optional>

<ref name="draw-image-map"/>

</optional>

<optional>

<ref name="svg-title"/>

</optional>

<optional>

<ref name="svg-desc"/>

</optional>

<optional>

<choice>

<ref name="draw-contour-polygon"/>

<ref name="draw-contour-path"/>

</choice>

</optional>

</element>

</define>

The attributes that may be associated with the <draw:frame> element are:

* Position, Size (relative sizes, see below), Style, Layer, Z-Index, ID, Caption ID and Transformation – see section 9.2.15.
* Text anchor, table background, draw end position – see section 9.2.16
* Presentation class – see section 9.6.1
* Copy frames

The following elements may be contained in the image element:

* Event Listeners – see section 12.4.
* Glue Points – see section 9.2.19.
* Image Map – see section 9.3.10.
* Title (short accessible name) – see section 9.2.20.
* Long description (in support of accessibility) – see section 9.2.20.
* Contour – see section 9.3.8.

#### Relative Sizes

For frames, the width and height of the drawing object may be specified as a relative value using the style:rel-width and style:rel-height attributes. The relative value either is a percentage value, the special value scale, or the special value scale-min.

The interpretation of relative values depends on the anchor of the drawing object. If the anchor for the drawing object is in a table cell, the percentage value relates to the surrounding table box. If the anchor for the drawing object is in a text box, the percentage value relates to the surrounding text box. In other cases, the percentage values relate to the width of the page or window.

The value scale for the width means that the width should be calculated depending on the height, so that the ratio of with and height of the original image or object size is preserved.

The value scale for the height means that the height should be calculated depending on the width, so that the ratio of with and height of the original image or object size is preserved.

The value scale-min equals the value scale, except that the calculated width or height is a minimum height rather than an absolute one.

To support application that don't support relative with and heights, applications that save the attributes style:rel-width or style:rel-height should also provide the real width and heights in the svg:width and svg:height/fo:min-height attributes.

<define name="common-draw-rel-size-attlist">

<ref name="common-draw-size-attlist"/>

<optional>

<attribute name="style:rel-width">

<choice>

<ref name="percent"/>

<value>scale</value>

<value>scale-min</value>

</choice>

</attribute>

</optional>

<optional>

<attribute name="style:rel-height">

<choice>

<ref name="percent"/>

<value>scale</value>

<value>scale-min</value>

</choice>

</attribute>

</optional>

</define>

#### Copy Frames

Multiple frames can be set to display the exact same underlying data: for instance for a company logo, that must appear somewhere on every page, without being part of a header or footer.

A frame can be set to display the contents of another frame, referenced by the draw:copy-of attribute. This does not effect style and position information. This is, the frame that has the draw:copy-of attribute has its own style and position information and does not use the one of the referenced frame.

<define name="draw-frame-attlist" combine="interleave">

<optional>

<attribute name="draw:copy-of">

<ref name="string"/>

</attribute>

</optional>

</define>

### Text Box

The <draw:text-box>element represents a text box. A text box may be used to place text in a container that is outside of the normal flow of the document.

<define name="draw-text-box">

<element name="draw:text-box">

<ref name="draw-text-box-attlist"/>

<zeroOrMore>

<ref name="text-content"/>

</zeroOrMore>

</element>

</define>

The attributes that may be associated with the <draw:text-box> element are:

* Chain
* Round Corners
* Minimum Height and Width
* Maximum Height and Width

Text boxes don't support contours as described in section 9.3.8 and alternative texts as described in section 9.2.20.

#### Chain

Text boxes can be chained, in other words, if the content of a text box exceeds its capacity, the content flows into the next text box in the chain. To chain text boxes, the attribute draw:chain-next-name is used, The value of this attribute is the name of the next text box in the chain. Chained text boxes usually are supported by text documents only.

<define name="draw-text-box-attlist" combine="interleave">

<optional>

<attribute name="draw:chain-next-name">

<ref name="string"/>

</attribute>

</optional>

</define>

#### Round Corners

The attribute draw:corner-radius specifies the radius of the circle used to round off the corners of the text-box.

<define name="draw-text-box-attlist" combine="interleave">

<optional>

<attribute name="draw:corner-radius">

<ref name="nonNegativeLength"/>

</attribute>

</optional>

</define>

#### Minimum Height and Width

The fo:min-height and fo:min-width attributes specify a minimum height or width for a text box. If they are existing, they overwrite the height or width of a text box specified by the svg:height and svg:width attributes of the surrounding <draw:frame> element. Their value can be either a length or a percentage. If the anchor for the text box is in a table cell, the percentage value relates to the surrounding table box. If the anchor for the text box is in a text box, the percentage value relates to the surrounding text box. In other cases, the percentage values relate to the height of the page or window.

<define name="draw-text-box-attlist" combine="interleave">

<optional>

<attribute name="fo:min-height">

<choice>

<ref name="length"/>

<ref name="percent"/>

</choice>

</attribute>

</optional>

<optional>

<attribute name="fo:min-width">

<choice>

<ref name="length"/>

<ref name="percent"/>

</choice>

</attribute>

</optional>

</define>

#### Maximum Height and Width

If the width or height of a text box is specified as a minimum width or height (using the fo:min-width or fo:min-height attributes), then the fo:max-width and fo:max-height attributes specify a maximum width and height for the text box. When these maximum values are reached, the text box stops increasing in size. The attributes' value can be either a length or a percentage. If the anchor for the text box is in a table cell, the percentage value relates to the size of the surrounding table cell. If the anchor for the text box is in a text box, the percentage value relates to the size of the surrounding text box. In other cases, the percentage values relate to the width or height of the page or window.

<define name="draw-text-box-attlist" combine="interleave">

<optional>

<attribute name="fo:max-height">

<choice>

<ref name="length"/>

<ref name="percent"/>

</choice>

</attribute>

</optional>

<optional>

<attribute name="fo:max-width">

<choice>

<ref name="length"/>

<ref name="percent"/>

</choice>

</attribute>

</optional>

</define>

#### ID

A text box may have an ID. This ID can be used to reference the text box from other elements.

<define name="draw-text-box-attlist" combine="interleave">

<optional>

<ref name="text-id"/>

</optional>

</define>

### Image

The <draw:image> element represents an image. An image can be either:

* Contained in a document as a link to an external resource

or

* Embedded in a document

This element can be an [XLink], in which case the element contains some attributes with fixed values that describe the link semantics.

While the image data may have an arbitrary format, it is recommended that vector graphics are stored in the [SVG] format and bitmap graphics in the [PNG] format.

<define name="draw-image">

<element name="draw:image">

<ref name="draw-image-attlist"/>

<choice>

<ref name="common-draw-data-attlist"/>

<ref name="office-binary-data"/>

</choice>

<ref name="draw-text"/>

</element>

</define>

The attributes that may be associated with the <draw:image> element are:

* Image data
* Filter name

Like most other drawing shapes, image drawing shapes may have text content. It is displayed in addition to the image data.

#### Image Data

The image data can be stored in one of the following ways:

* The image data is contained in an external file. Use the xlink:href and associated attributes described below to link to the external file.
* The image data is contained in the <draw:image> element. The <draw:image> then element contains an <office:binary-data> element that contains the image data in BASE64 encoding (as defined in [RFC2045]). In this situation the xlink:href attribute is not required.

<define name="common-draw-data-attlist" combine="interleave">

<group>

<attribute name="xlink:href">

<ref name="anyURI"/>

</attribute>

<optional>

<attribute name="xlink:type" a:defaultValue="simple">

<choice>

<value>simple</value>

</choice>

</attribute>

</optional>

<optional>

<attribute name="xlink:show" a:defaultValue="embed">

<choice>

<value>embed</value>

</choice>

</attribute>

</optional>

<optional>

<attribute name="xlink:actuate" a:defaultValue="onLoad">

<choice>

<value>onLoad</value>

</choice>

</attribute>

</optional>

</group>

</define>

<define name="office-binary-data">

<element name="office:binary-data">

<ref name="base64Binary"/>

</element>

</define>

#### Filter Name

If required, the draw:filter-name attribute can represent the filter name of the image. This attribute contains the internal filter name that the office application software used to load the graphic.

<define name="draw-image-attlist" combine="interleave">

<optional>

<attribute name="draw:filter-name">

<ref name="string"/>

</attribute>

</optional>

</define>

### Objects

A document in OpenDocument format can contain two types of objects, as follows:

* Objects that have an OpenDocument or other XML representation. Objects that have an OpenDocument representation are:
  + Formulas (represented as [MathML])
  + Charts
  + Spreadsheets
  + Text documents
  + Drawings
  + Presentations
* Objects that do not have an XML representation. These objects only have a binary representation, An example for this kind of objects OLE objects (see [OLE]).

The <draw:object> element represents objects that have a XML representation. The <draw:object-ole> element represents objects that only have a binary representation.

<define name="draw-object">

<element name="draw:object">

<ref name="draw-object-attlist"/>

<choice>

<ref name="common-draw-data-attlist"/>

<ref name="office-document"/>

<ref name="math-math"/>

</choice>

</element>

</define>

<define name="draw-object-ole">

<element name="draw:object-ole">

<ref name="draw-object-ole-attlist"/>

<choice>

<ref name="common-draw-data-attlist"/>

<ref name="office-binary-data"/>

</choice>

</element>

</define>

The attributes that may be associated with the <draw:object> and <draw:object-ole> elements are:

* Object data
* Table Change Notifications
* Class Id

Objects do not support transformations as described in section 9.2.15.

#### Object Data

The object data can be called in one of the following ways:

* The xlink:href attribute links to the object representation, as follows:
  + For objects that have an XML representation, the link references the sub package of the object. The object is contained within this sub page exactly as it would as it is a document of its own.
  + For objects that do not have an XML representation, the link references a sub stream of the package that contains the binary representation of the object.

Application that support objects should support linking to objects that are contained within the same package. They may also support linking to object located outside the package.

* The object data is contained in the <draw:object> or <draw:object-ole> element, as follows:
  + The <draw:object> element contains the XML representation of the object, for example, an <office:document> or a <math:math> element.
  + The <draw:object-ole> element contains an <office:binary-data> element, which contains the binary data for the object in BASE64 encoding.

In these situations, the xlink:href attributes are not required.

The xlink:href attribute is described in section 9.3.2.

It is recommended to include an image representation of the object into the frame in addition to the object itself.

#### Notification on Table Change

Some objects, especially charts, may require a notification when a table in the document changes. To enable this notification, use the draw:notify-on-change-of-table attribute, which contains the name of the table. This attribute can be associated with the <draw:object> element.

<define name="draw-object-attlist" combine="interleave">

<optional>

<attribute name="draw:notify-on-update-of-ranges">

<ref name="string"/>

</attribute>

</optional>

</define>

#### Class Id

If the embedded object is an OLE object, the draw:class-id attribute optionally contains the OLE class id of the object (see also [OLE]).

<define name="draw-object-ole-attlist" combine="interleave">

<optional>

<attribute name="draw:class-id"/>

</optional>

</define>

### Applet

An applet is a small Java-based program that is embedded in a document. The <draw:applet> element is based on the <applet> tag in [HTML4]. This element must contain either the draw:code or draw:object attribute.

<define name="draw-applet">

<element name="draw:applet">

<ref name="draw-applet-attlist"/>

<optional>

<ref name="common-draw-data-attlist"/>

</optional>

<zeroOrMore>

<ref name="draw-param"/>

</zeroOrMore>

</element>

</define>

The attributes that may be associated with the <draw:applet> element are:

* Codebase
* Code
* Object
* Archive
* Mayscript

The only element that may be contained in the <draw:applet> element is:

* Parameter (see section 9.3.6)

Applets do not support transformations as described in section 9.2.15.

#### Codebase

The codebase specifies the base IRI for the applet. If this attribute is not specified, then it defaults the same base IRI as for the current document. The codebase is represented be the [XLink] attributes xlink:href, xlink:type, xlink:show, and xlink:actuate. The xlink:href attribute is described in section 9.3.2.

#### Code

The draw:code attribute specifies one of the following:

* The name of the class file that contains the compiled applet subclass.
* The path to the class, including the class file itself.

Either this attribute or the draw:object attribute is required. The value of this attribute is interpreted in relation to the codebase for the applet.

<define name="draw-applet-attlist" combine="interleave">

<optional>

<attribute name="draw:code"/>

</optional>

</define>

#### Object

The draw:object attribute specifies a resource that contains a serialized representation of the state of the applet. The serialized data contains the class name of the applet but not the implementation. The value of this attribute is interpreted in relation to the codebase for the applet.

<define name="draw-applet-attlist" combine="interleave">

<optional>

<attribute name="draw:object"/>

</optional>

</define>

#### Archive

The draw:archive attribute specifies a comma-separated list of URLs for archives that contain classes and other resources that are preloaded.

<define name="draw-applet-attlist" combine="interleave">

<optional>

<attribute name="draw:archive"/>

</optional>

</define>

#### Mayscript

The draw:mayscript attribute specifies whether or not the applet can be scripted.

<define name="draw-applet-attlist" combine="interleave">

<optional>

<attribute name="draw:may-script" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

### Plugins

A plugin is a binary object that is plugged into a document to represent a media-type that usually is not handled natively by office application software. Plugins are represented by the <draw:plugin> element

<define name="draw-plugin">

<element name="draw:plugin">

<ref name="draw-plugin-attlist"/>

<ref name="common-draw-data-attlist"/>

<zeroOrMore>

<ref name="draw-param"/>

</zeroOrMore>

</element>

</define>

The attributes that may be associated with the <draw:plugin> element are:

* Mime type
* Source

The only element that may be contained in the <draw:plugin> element is:

* Parameter (see section 9.3.6)

Plugins do not support transformations as described in section 9.2.15.

#### Mime type

The draw:mimetype attribute specifies the MIME type to which this plugin should be registered.

<define name="draw-plugin-attlist" combine="interleave">

<optional>

<attribute name="draw:mime-type"/>

</optional>

</define>

#### Source

The [XLink] attributes xlink:href, xlink:type, xlink:show, and xlink:actuate specify the source of the plugin. The xlink:href attribute is described in section 9.3.2.

### Parameters

The <draw:param> element contains parameters that are passed to an applet or plugin when they are initialized.

<define name="draw-param">

<element name="draw:param">

<ref name="draw-param-attlist"/>

<empty/>

</element>

</define>

The attributes that may be associated with the <draw:param> element are:

* Name
* Value

#### Name

The draw:name attribute specifies the name of a runtime parameter.

<define name="draw-param-attlist" combine="interleave">

<optional>

<attribute name="draw:name"/>

</optional>

</define>

#### Value

The draw:value attribute specifies the value of the runtime parameter specified by the name.

<define name="draw-param-attlist" combine="interleave">

<optional>

<attribute name="draw:value"/>

</optional>

</define>

### Floating Frame

A floating frame is a frame embedded in a document, which may contain, for example, a text document or spreadsheet. A floating frame is represented by the <draw:floating-frame> element.

<define name="draw-floating-frame">

<element name="draw:floating-frame">

<ref name="draw-floating-frame-attlist"/>

<ref name="common-draw-data-attlist"/>

</element>

</define>

The attributes that may be associated with the <draw:floating-frame> element are:

* Source
* Frame Name

Floating frames do not support transformations as described in section 9.2.15.

#### Source

The [XLink] attributes xlink:href, xlink:type, xlink:show, and xlink:actuate specify the source of the floating frame. The xlink:href attribute is described in section 9.3.2.

#### Frame Name

The draw:frame-name specifies the name of the frame. This name can be used as target from within hyperlinks.

<define name="draw-floating-frame-attlist" combine="interleave">

<optional>

<attribute name="draw:frame-name">

<ref name="string"/>

</attribute>

</optional>

</define>

### Contour

The <draw:contour-polygon> and <draw:contour-path> elements may be contained in the following elements:

* <draw:image>
* <draw:object>
* <draw:object-ole>
* <draw:applet>
* <draw:plugin>
* <draw:floating-frame>

These elements describe the contour of an image or object.

<define name="draw-contour-polygon">

<element name="draw:contour-polygon">

<ref name="common-contour-attlist"/>

<ref name="common-draw-size-attlist"/>

<ref name="common-draw-viewbox-attlist"/>

<ref name="common-draw-points-attlist"/>

<empty/>

</element>

</define>

<define name="draw-contour-path">

<element name="draw:contour-path">

<ref name="common-contour-attlist"/>

<ref name="common-draw-size-attlist"/>

<ref name="common-draw-viewbox-attlist"/>

<ref name="common-draw-path-data-attlist"/>

<empty/>

</element>

</define>

The elements are similar to the <draw:polygon> (see section 9.2.4) and <draw:path> (see section 9.2.6) elements, except that they specify a contour rather than a drawing shape. The attributes they support are the ones for the size, the viewbox, the points (contour polygon only) and the path (contour path only).

For the svg:width and svg:height attributes of the <draw:contour-polygon> and <draw:contour-path> elements, applications should support pixel lengths (i.e., 20px) in addition to traditional lengths like 2cm.

#### Recreate on Edit

The draw:recreate-on-edit attribute specifies if the contour of the image or object should be recreated automatically when the image or object is edited.

<define name="common-contour-attlist" combine="interleave">

<attribute name="draw:recreate-on-edit">

<ref name="boolean"/>

</attribute>

</define>

### Hyperlinks

Frames may behave like hyperlinks. Such hyperlinks are represented by the <draw:a> element, where. the element's content is the frame that should be the source of the link.

This element is an [XLink] and has some attributes with fixed values and describe the semantics of the link.

<define name="draw-a">

<element name="draw:a">

<ref name="draw-a-attlist"/>

<ref name="draw-frame"/>

</element>

</define>

The attributes that may be associated with the <draw:a> element are:

* Link location
* Link target frame
* Name
* Title
* Server side image map

#### Link Location

The xlink:href attribute specifies the target location of the link.

<define name="draw-a-attlist" combine="interleave">

<attribute name="xlink:href">

<ref name="anyURI"/>

</attribute>

<optional>

<attribute name="xlink:type" a:defaultValue="simple">

<value>simple</value>

</attribute>

</optional>

<optional>

<attribute name="xlink:actuate" a:defaultValue="onRequest">

<choice>

<value>onRequest</value>

</choice>

</attribute>

</optional>

</define>

#### Link Target Frame

The office:target-frame attribute specifies the target frame of the link.

This attribute can have one of the following values:

* \_self : The referenced document replaces the content of the current frame.
* \_blank : The referenced document is displayed in a new frame.
* \_parent : The referenced document is displayed in the parent frame of the current frame.
* \_top : The referenced document is displayed in the topmost frame, that is the frame that contains the current frame as a child or descendent but is not contained within another frame.
* A frame name : The referenced document is displayed in the named frame. If the named frame does not exist, a new frame with that name is created.

To conform with the [XLink] specification, an additional xlink:show attribute is attached to the <draw:a> element. If the value of the this attribute is \_blank, the xlink:show attribute value is new. If the value of the this attribute is any of the other value options, the value of the xlink:show attribute is replace.

<define name="draw-a-attlist" combine="interleave">

<optional>

<attribute name="office:target-frame-name">

<ref name="targetFrameName"/>

</attribute>

</optional>

<optional>

<attribute name="xlink:show">

<choice>

<value>new</value>

<value>replace</value>

</choice>

</attribute>

</optional>

</define>

#### Name

A hyperlink can have a name, but it is not essential. The office:name attribute specifies the name of the link. The name can serve as a target for other hyperlinks. The name does not have to be unique.

This attribute is specified for compatibility with [HTML4] only, where an <a> element may serve as a link source and target simultaneously. We strongly recommend that this attribute not be used for any purpose other than to represent links that originally came from a HTML document.

<define name="draw-a-attlist" combine="interleave">

<optional>

<attribute name="office:name">

<ref name="string"/>

</attribute>

</optional>

</define>

#### Title

The office:title attribute specifies a short accessible description for hint text.

See appendix E for guidelines how to use this attribute.

<define name="draw-a-attlist" combine="interleave">

<optional>

<attribute name="office:title">

<ref name="string"/>

</attribute>

</optional>

</define>

#### Server Side Image Map

A link can be a server side image map. If the office:server-map attribute is present, the mouse coordinates of the click position of the graphic shape are appended to the IRI of the link. The coordinates may be used by the server to determine which link to activate within the image map.

<define name="draw-a-attlist" combine="interleave">

<optional>

<attribute name="office:server-map" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

### Client Side Image Maps

An client side image map is a collection of hyperlinks that are associated with graphic elements. The image map is a sequence of image map elements. Each image map element associates a hyperlink with an area. The area can be one of the following shapes:

* Rectangular
* Circular
* Polygonal

The <draw:image-map> element represents an image map.

<define name="draw-image-map">

<element name="draw:image-map">

<zeroOrMore>

<choice>

<ref name="draw-area-rectangle"/>

<ref name="draw-area-circle"/>

<ref name="draw-area-polygon"/>

</choice>

</zeroOrMore>

</element>

</define>

The <draw:image-map> element can contain three types of image map elements, which represent the three types of image map areas as follows:

* Rectangular image map elements
* Circular image map elements
* Polygonal image map elements

Image map elements are described in terms of absolute positions. When loading the XML file, the office application must map the image map onto its associated graphical element, for example an image, in its original size. The application then must scale the image map to match the current size of the image, but in the file format the image is always saved in its unscaled version, matching the dimensions of the unscaled image.

#### Rectangular Image Map Areas

The <draw:area-rectangle> element describes a rectangular image map area by an x, y position (svg:x and svg:y attributes) as well as a width and the height (svg:width and svg:height attributes). These attributes are required. In addition to this, the attributes described in section 9.3.10:Common Image Map Attributes and Elements are optionally supported.

<define name="draw-area-rectangle">

<element name="draw:area-rectangle">

<ref name="common-draw-area-attlist"/>

<attribute name="svg:x">

<ref name="coordinate"/>

</attribute>

<attribute name="svg:y">

<ref name="coordinate"/>

</attribute>

<attribute name="svg:width">

<ref name="length"/>

</attribute>

<attribute name="svg:height">

<ref name="length"/>

</attribute>

<optional>

<ref name="svg-title"/>

</optional>

<optional>

<ref name="svg-desc"/>

</optional>

<optional>

<ref name="office-event-listeners"/>

</optional>

</element>

</define>

#### Circular Image Map Areas

The <draw:area-circle> element describes a circular image map area. The additional attributes for circular image maps are described below in the common attributes section.

The required attributes svg:cx and svg:cy specify the center point of the circle. The required svg:r attribute specifies the radius of the circle.

The attributes described in section 9.3.10:Common Image Map Attributes and Elements are optional.

<define name="draw-area-circle">

<element name="draw:area-circle">

<ref name="common-draw-area-attlist"/>

<attribute name="svg:cx">

<ref name="coordinate"/>

</attribute>

<attribute name="svg:cy">

<ref name="coordinate"/>

</attribute>

<attribute name="svg:r">

<ref name="length"/>

</attribute>

<optional>

<ref name="svg-title"/>

</optional>

<optional>

<ref name="svg-desc"/>

</optional>

<optional>

<ref name="office-event-listeners"/>

</optional>

</element>

</define>

#### Polygonal Image Map Areas

The <draw:area-polygon> element describes a polygonal image map area. A polygonal image map area is comprised of the following components:

* A bounding box.  
  The bounding box, which is represented in the same way as a rectangular image map area using the svg:x, svg:y, svg:width, and svg:height attributes, establishes the reference frame for the view box and the polygon point sequence. The reference frame enables the coordinates to be translated into absolute coordinates.
* A view box.  
  The view box attribute svg:viewBox establishes a coordinate system for the point sequence. The view box obviates the need to record every point of the point sequence as absolute coordinates with length and unit of measurement.
* A sequence of points in view box coordinates in the svg:points attribute.

For more information about how to represent polygons, see section 9.2.4.

The attributes above are required. The attributes described in section 9.3.10:Common Image Map Attributes and Elements are optional.

<define name="draw-area-polygon">

<element name="draw:area-polygon">

<ref name="common-draw-area-attlist"/>

<attribute name="svg:x">

<ref name="coordinate"/>

</attribute>

<attribute name="svg:y">

<ref name="coordinate"/>

</attribute>

<attribute name="svg:width">

<ref name="length"/>

</attribute>

<attribute name="svg:height">

<ref name="length"/>

</attribute>

<ref name="common-draw-viewbox-attlist"/>

<ref name="common-draw-points-attlist"/>

<optional>

<ref name="svg-title"/>

</optional>

<optional>

<ref name="svg-desc"/>

</optional>

<optional>

<ref name="office-event-listeners"/>

</optional>

</element>

</define>

Example: Polygonal image map area

The element shown in the following example defines a triangle that is located in the middle of a 2cm by 2cm image. The bounding box covers an area of 2cm by 1.5cm. One view box unit corresponds to 0.01mm.

<draw:area-polygon ...   
 svg:x="0" svg:y="0" svg:width="2.0cm" svg:height="2.0cm"  
 svg:viewBox="0 0 2000 2000"  
 svg:points="400,1500 1600,1500 1000,400"/>

#### Common Image Map Attributes and Elements

In addition to the shape attributes, each image map element can contain the following information:

* Link, including a IRI and link target frame.
* Name.
* Inactive flag.
* Title (short accessible name). Use the <svg:title> child element as described in section 9.2.20.
* Long description (in support of accessibility). Use the <svg:desc> child element as described in section 9.2.20.
* Events associated with the area. Use the <office:event-listeners> child element as described in section 12.4.

Other attributes of the image maps are taken from the HTML image map representation.

Each image map element identifies a hyperlink and uses the [XLink] href, type, and show attributes, and the office:target-frame-name attribute to describe the link.

<define name="common-draw-area-attlist" combine="interleave">

<optional>

<attribute name="xlink:href">

<ref name="anyURI"/>

</attribute>

</optional>

<optional>

<attribute name="xlink:type" a:defaultValue="simple">

<choice>

<value>simple</value>

</choice>

</attribute>

</optional>

<optional>

<attribute name="office:target-frame-name">

<ref name="targetFrameName"/>

</attribute>

</optional>

<optional>

<attribute name="xlink:show">

<choice>

<value>new</value>

<value>replace</value>

</choice>

</attribute>

</optional>

</define>

The office:name attribute assigns a name to each image map element.

<define name="common-draw-area-attlist" combine="interleave">

<optional>

<attribute name="office:name">

<ref name="string"/>

</attribute>

</optional>

</define>

The draw:nohref attribute declares that the image map element and the associated area is inactive. The IRI that is contained in the image map element is not used.

<define name="common-draw-area-attlist" combine="interleave">

<optional>

<attribute name="draw:nohref">

<choice>

<value>nohref</value>

</choice>

</attribute>

</optional>

</define>

## 3D Shapes

### Scene

The <dr3d:scene> element is the only element that can contain three-dimensional shapes. A scene is like a group, but it also defines the projection, lighting, and other render details for the shapes inside the scene.

<define name="dr3d-scene">

<element name="dr3d:scene">

<ref name="dr3d-scene-attlist"/>

<ref name="common-draw-position-attlist"/>

<ref name="common-draw-size-attlist"/>

<ref name="common-draw-style-name-attlist"/>

<ref name="common-draw-z-index-attlist"/>

<ref name="common-draw-id-attlist"/>

<ref name="common-draw-layer-name-attlist"/>

<ref name="common-text-spreadsheet-shape-attlist"/>

<ref name="common-dr3d-transform-attlist"/>

<ref name="common-draw-caption-id-attlist"/>

<optional>

<ref name="svg-title"/>

</optional>

<optional>

<ref name="svg-desc"/>

</optional>

<zeroOrMore>

<ref name="dr3d-light"/>

</zeroOrMore>

<zeroOrMore>

<ref name="shapes3d"/>

</zeroOrMore>

</element>

</define>

<define name="shapes3d">

<choice>

<ref name="dr3d-scene"/>

<ref name="dr3d-extrude"/>

<ref name="dr3d-sphere"/>

<ref name="dr3d-rotate"/>

<ref name="dr3d-cube"/>

</choice>

</define>

The attributes that may be associated with the <dr3d:scene> element are:

* Position, Size, Style, Layer, Z-Index, ID and Caption ID – see section 9.2.15
* Text anchor, table background, draw end position – see section 9.2.16
* Camera vectors
* Projection
* Distance
* Focal length
* Shadow slant
* Shade mode
* Ambient color
* Lighting mode

The elements that may be contained in the <dr3d:scene> element are:

* Title (short accessible name) – see section 9.2.20.
* Long description (in support of accessibility) – see section 9.2.20.
* Light – see section 9.4.2.
* Scene – see section 9.4.1.
* Extrude – see section 9.4.5.
* Sphere – see section 9.4.4.
* Rotate – see section 9.4.6.
* Cube – see section 9.4.3.

#### Camera Vectors

The camera vectors define a viewing volume. The dr3d:vrp attribute specifies the origin, the dr3d:vpn attribute points towards the projected objects, and the dr3d:vup attribute defines the up vector.

<define name="dr3d-scene-attlist" combine="interleave">

<optional>

<attribute name="dr3d:vrp">

<ref name="vector3D"/>

</attribute>

</optional>

<optional>

<attribute name="dr3d:vpn">

<ref name="vector3D"/>

</attribute>

</optional>

<optional>

<attribute name="dr3d:vup">

<ref name="vector3D"/>

</attribute>

</optional>

</define>

#### Projection

The dr3d:projection attribute specifies the projection. The projection can be perspective or parallel. In perspective mode, objects become smaller in the distance.

<define name="dr3d-scene-attlist" combine="interleave">

<optional>

<attribute name="dr3d:projection">

<choice>

<value>parallel</value>

<value>perspective</value>

</choice>

</attribute>

</optional>

</define>

#### Distance

The dr3d:distance attribute specifies the distance between the camera and the object.

<define name="dr3d-scene-attlist" combine="interleave">

<optional>

<attribute name="dr3d:distance">

<ref name="length"/>

</attribute>

</optional>

</define>

#### Focal Length

The dr3d:focal-length attribute specifies the length of the focus for the virtual camera of this scene.

<define name="dr3d-scene-attlist" combine="interleave">

<optional>

<attribute name="dr3d:focal-length">

<ref name="length"/>

</attribute>

</optional>

</define>

#### Shadow Slant

The dr3d:shadow-slant attribute defines the angle from the three-dimensional scene to a virtual paper on which the shadow is cast.

<define name="dr3d-scene-attlist" combine="interleave">

<optional>

<attribute name="dr3d:shadow-slant">

<ref name="nonNegativeInteger"/>

</attribute>

</optional>

</define>

#### Shade Mode

The shade mode defines how the lighting is calculated for rendered surfaces

* flat: lighting is calculated by one surface normal.
* phong: lighting is calculated by interpolating the surface normals over the surface.
* gouraud: lighting is calculated by interpolating the color calculated with the surface normals at each edge.
* draft: surfaces are not lit and drawn as wireframe only.

<define name="dr3d-scene-attlist" combine="interleave">

<optional>

<attribute name="dr3d:shade-mode">

<choice>

<value>flat</value>

<value>phong</value>

<value>gouraud</value>

<value>draft</value>

</choice>

</attribute>

</optional>

</define>

#### Ambient Color

The dr3d:ambient-color attribute specifies the color for ambient light. Ambient light is that light that seems to come from all directions.

<define name="dr3d-scene-attlist" combine="interleave">

<optional>

<attribute name="dr3d:ambient-color">

<ref name="color"/>

</attribute>

</optional>

</define>

#### Lighting Mode

The attribute dr3d:lighting-mode enables or disables the use of lighting in the three-dimensional scene.

<define name="dr3d-scene-attlist" combine="interleave">

<optional>

<attribute name="dr3d:lighting-mode">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### 3D Transformation

The value of the dr3d:transform attribute is a list of transform definitions, which are applied in the order provided. The individual transform definitions are separated by whitespace. The available types of transform definitions include:

* matrix (<a> <b> <c> <d> <e> <f> <g> <h> <i> <j> <k> <l>), which specifies a transformation in the form of a transformation matrix of six values. matrix(a,b,c,d,e,f,g,h,i,j,k,l) is equivalent to applying the transformation matrix [a b c d e f g h i j k l].
* translate (<tx> <ty> <tz>), which specifies a translation by tx, ty and tz.
* scale (<sx> <sy> <sz>), which specifies a scale operation by sx, sy and sz.
* rotatex (<rotate-angle> ), which specifies a rotation by <rotate-angle> degrees along the x-axis.
* rotatey (<rotate-angle> ), which specifies a rotation by <rotate-angle> degrees along the y-axis.
* rotatez (<rotate-angle> ), which specifies a rotation by <rotate-angle> degrees along the y-axis.

<define name="common-dr3d-transform-attlist">

<optional>

<attribute name="dr3d:transform"/>

</optional>

</define>

### Light

The <dr3d:light> element represents a light inside a scene.

This element must be the first element contained in a <dr3d:scene> element. There may be several lights, but applications may only support a limited number per scene. A typical limitation are 8 lights per scene.

<define name="dr3d-light">

<element name="dr3d:light">

<ref name="dr3d-light-attlist"/>

<empty/>

</element>

</define>

The attributes that may be associated with the <dr3d:light> element are:

* Diffuse color
* Direction
* Enabled
* Specular

#### Diffuse Color

The dr3d:diffuse-color attribute specifies the base color that the light is emitting.

<define name="dr3d-light-attlist" combine="interleave">

<optional>

<attribute name="dr3d:diffuse-color">

<ref name="color"/>

</attribute>

</optional>

</define>

#### Direction

The dr3d:direction attribute specifies the direction in which the light is emitted.

<define name="dr3d-light-attlist" combine="interleave">

<attribute name="dr3d:direction">

<ref name="vector3D"/>

</attribute>

</define>

#### Enabled

The dr3d:enabled attribute specifies whether or not the light is enabled. If a light is not enabled, it does not emit any light.

<define name="dr3d-light-attlist" combine="interleave">

<optional>

<attribute name="dr3d:enabled">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Specular

The dr3d:specular attribute specifies whether or not the light causes a specular reflection on the objects. Applications may evaluate this attribute only for the first light in a scene.

<define name="dr3d-light-attlist" combine="interleave">

<optional>

<attribute name="dr3d:specular">

<ref name="boolean"/>

</attribute>

</optional>

</define>

### Cube

The <dr3d:cube> element represents a three-dimensional cube shape.

<define name="dr3d-cube">

<element name="dr3d:cube">

<ref name="dr3d-cube-attlist"/>

<ref name="common-draw-z-index-attlist"/>

<ref name="common-draw-id-attlist"/>

<ref name="common-draw-layer-name-attlist"/>

<ref name="common-draw-style-name-attlist"/>

<ref name="common-dr3d-transform-attlist"/>

<empty/>

</element>

</define>

The attributes that may be associated with the <dr3d:cube> element are:

* Style, Layer, Z-Index and ID – see section 9.2.15
* Minimum and Maximum Edge

#### Minimum and Maximum Edge

The attributes dr3d:min-edge and dr3d:max-edge specify the minimum and maximum edge of the cube in a 3D space.

<define name="dr3d-cube-attlist" combine="interleave">

<optional>

<attribute name="dr3d:min-edge">

<ref name="vector3D"/>

</attribute>

</optional>

<optional>

<attribute name="dr3d:max-edge">

<ref name="vector3D"/>

</attribute>

</optional>

</define>

### Sphere

The <dr3d:sphere> element represents a three-dimensional sphere shape.

<define name="dr3d-sphere">

<element name="dr3d:sphere">

<ref name="dr3d-sphere-attlist"/>

<ref name="common-draw-z-index-attlist"/>

<ref name="common-draw-id-attlist"/>

<ref name="common-draw-layer-name-attlist"/>

<ref name="common-draw-style-name-attlist"/>

<ref name="common-dr3d-transform-attlist"/>

<empty/>

</element>

</define>

The attributes that may be associated with the <dr3d:sphere> element are:

* Style, Layer, Z-Index, and ID – see section 9.2.15
* Center
* Size

#### Center

The dr3d:center attribute defines the center of the sphere in a three-dimensional space.

<define name="dr3d-sphere-attlist" combine="interleave">

<optional>

<attribute name="dr3d:center">

<ref name="vector3D"/>

</attribute>

</optional>

</define>

#### Size

The dr3d:size attribute defines the size of the sphere in a three-dimensional space.

<define name="dr3d-sphere-attlist" combine="interleave">

<optional>

<attribute name="dr3d:size">

<ref name="vector3D"/>

</attribute>

</optional>

</define>

### Extrude

The <dr3d:extrude> element represents a three-dimensional extrude based on a polygon.

<define name="dr3d-extrude">

<element name="dr3d:extrude">

<ref name="common-draw-path-data-attlist"/>

<ref name="common-draw-viewbox-attlist"/>

<ref name="common-draw-id-attlist"/>

<ref name="common-draw-z-index-attlist"/>

<ref name="common-draw-layer-name-attlist"/>

<ref name="common-draw-style-name-attlist"/>

<ref name="common-dr3d-transform-attlist"/>

<empty/>

</element>

</define>

The attributes that may be associated with the <dr3d:extrude> element are:

* Viewbox, Style, Layer, Z-Index, and ID – see section 9.2.15
* Path Data – see section 9.2.6

### Rotate

The <dr3d:rotate> element represents a three-dimensional rotation shape based on a polygon.

<define name="dr3d-rotate">

<element name="dr3d:rotate">

<ref name="common-draw-viewbox-attlist"/>

<ref name="common-draw-path-data-attlist"/>

<ref name="common-draw-z-index-attlist"/>

<ref name="common-draw-id-attlist"/>

<ref name="common-draw-layer-name-attlist"/>

<ref name="common-draw-style-name-attlist"/>

<ref name="common-dr3d-transform-attlist"/>

<empty/>

</element>

</define>

The attributes that may be associated with the <dr3d:rotate> element are:

* Viewbox, Style, Layer, Z-Index, and ID – see section 9.2.15
* Path Data – see section 9.2.6

## Custom Shape

A <draw:custom-shape> represents a shape that is capable of rendering complex figures. It is offering font work and extrusion functionality. A custom shape may have a geometry that influences its shape. This geometry may be visualized in office application user interfaces, for instance by displaying interaction handles, that provide a simple way to modify the the geometry.

<define name="draw-custom-shape">

<element name="draw:custom-shape">

<ref name="draw-custom-shape-attlist"/>

<ref name="common-draw-position-attlist"/>

<ref name="common-draw-size-attlist"/>

<ref name="common-draw-shape-with-text-and-styles-attlist"/>

<ref name="common-draw-caption-id-attlist"/>

<optional>

<ref name="svg-title"/>

</optional>

<optional>

<ref name="svg-desc"/>

</optional>

<optional>

<ref name="office-event-listeners"/>

</optional>

<zeroOrMore>

<ref name="draw-glue-point"/>

</zeroOrMore>

<ref name="draw-text"/>

<optional>

<ref name="draw-enhanced-geometry"/>

</optional>

</element>

</define>

The attributes that may be associated with the <draw:custom shape> element are:

* Position, Size, Style, Layer, Z-Index, ID, Caption ID and Transformation – see section 9.2.15.
* Text anchor, table background, draw end position – see section 9.2.16.
* Draw engine
* Draw data

The elements that may be contained in the <draw:custom-shape> element are:

* Title (short accessible name) – see section 9.2.20.
* Long description (in support of accessibility) – see section 9.2.20.
* Event listeners – see section 9.2.21.
* Glue points – see section 9.2.19.
* Text – see section 9.2.17.
* Enhanced geometry – see section 9.5.1,

#### Draw Engine

The optional draw:engine attribute specifies the name of a rendering engine that can be used to render the custom shape. The attribute's value is a namespaced token, meaning an identifier prefixed by an XML namespace prefix, just like any attribute or element name in this specification. The drawing engine may get its data either from the draw:data attribute, or it may evaluate the <draw:enhanced-geometry> child element.

If the draw:engine attribute is omitted, the office application's default enhanced custom shape rendering engine will be used. This engine gets its geometry data from the <draw:enhanced-geometry> element only.

<define name="draw-custom-shape-attlist" combine="interleave">

<optional>

<attribute name="draw:engine">

<ref name="namespacedToken"/>

</attribute>

</optional>

</define>

#### Draw Data

The draw:data attribute contains rendering engine specific data that describes the geometry of the custom shape. This attribute is only evaluated if a non default rendering engine is specified by the draw:engine attribute.

<define name="draw-custom-shape-attlist" combine="interleave">

<optional>

<attribute name="draw:data">

<ref name="string"/>

</attribute>

</optional>

</define>

### Enhanced Geometry

The <draw:enhanced-geometry> element contains the geometry for a <draw:custom-shape> element if its draw:engine attribute has been omitted.

<define name="draw-enhanced-geometry">

<element name="draw:enhanced-geometry">

<ref name="draw-enhanced-geometry-attlist"/>

<zeroOrMore>

<ref name="draw-equation"/>

</zeroOrMore>

<zeroOrMore>

<ref name="draw-handle"/>

</zeroOrMore>

</element>

</define>

The attributes that may be associated with the <draw:enhanced-geometry> element are

* Type
* View Box
* Mirror
* Text Rotate Angle
* Extrusion Allowed
* Text Path Allowed
* Concentric Gradient Fill Allowed
* Enhanced Geometry - Extrusion Attributes (see section 9.5.2)
* Enhanced Geometry - Path Attributes (see section 9.5.3)
* Enhanced Geometry - Text Path Attributes (see section 9.5.4)
* Enhanced Geometry - Equation (see section 9.5.5)
* Enhanced Geometry - Handle Attributes (see section 9.5.6)

#### Type

The draw:type attribute contains the name of a shape type. This name can be used to offer specialized user interfaces for certain classes of shapes, like for arrows, smileys, etc.

The shape type is rendering engine dependent and does not influence the geometry of the shape. If the value of the draw:type attribute is non-primitive, then no shape type is available.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="draw:type" a:defaultValue="non-primitive">

<ref name="custom-shape-type"/>

</attribute>

</optional>

</define>

<define name="custom-shape-type">

<choice>

<value>non-primitive</value>

<ref name="string"/>

</choice>

</define>

#### View Box

The svg:viewBox attribute establishes a user coordinate system inside the physical coordinate system of the shape specified by the position and size attributes. This user coordinate system is used by the <draw:enhanced-path> element.

The syntax for using this attribute is the same as the [SVG] syntax. The value of the attribute are four numbers separated by white spaces, which define the left, top, right, and bottom dimensions of the user coordinate system.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="svg:viewBox">

<list>

<ref name="integer"/>

<ref name="integer"/>

<ref name="integer"/>

<ref name="integer"/>

</list>

</attribute>

</optional>

</define>

#### Mirror

The draw:mirror-vertical and draw:mirror-horizontal attributes specify if the geometry of the shape is to be mirrored.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="draw:mirror-vertical" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

<optional>

<attribute name="draw:mirror-horizontal" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Text Rotate Angle

The draw:text-rotate-angle attribute specifies the angle by which the text within the custom shape is rotated in addition to the rotation included in the shape's draw:transform attribute.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="draw:text-rotate-angle" a:defaultValue="0">

<ref name="double"/>

</attribute>

</optional>

</define>

#### Extrusion Allowed

The draw:extrusion-allowed attribute specifies whether the shape is capable to be rendered as extrusion object.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="draw:extrusion-allowed" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Text Path Allowed

The draw:text-path-allowed attribute specifies if the shape is capable of being rendered as Fontwork object. The text of a Fontwork object is distinguished from normal text objects by being able to render text along or between lines that are specified by the draw:enhanced-path attribute. Fontwork objects are capable to support standard graphic attributes such as fill, shadow and or line styles.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="draw:text-path-allowed" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Concentric Gradient Fill Allowed

The draw:concentric-gradient-fill-allowed attribute specifies if the shape is capable being rendered with a concentric gradient that uses the custom shape path.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="draw:concentric-gradient-fill-allowed"

a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

### Enhanced Geometry - Extrusion Attributes

#### Extrusion

The draw:extrusion attribute determines if an extrusion is displayed.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="draw:extrusion" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Extrusion Brightness

The draw:extrusion-brightness attribute specifies the brightness of a scene.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="draw:extrusion-brightness" a:defaultValue="33%">

<ref name="percent"/>

</attribute>

</optional>

</define>

#### Extrusion Depth

The draw:extrusion-depth attribute specifies the depth of the extrusion. It takes two space separated values. The first value specifies the depth of the extrusion, the second value specifies the fraction of the extrusion that lies before the shape. It must be in the range [0,1]. A value of 0 is default.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="draw:extrusion-depth" a:defaultValue="36pt 0">

<list>

<ref name="length"/>

<ref name="double"/>

</list>

</attribute>

</optional>

</define>

#### Extrusion Diffusion

The amount of diffusion reflected by the shape is specified by the draw:extrusion-diffusion attribute.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="draw:extrusion-diffusion" a:defaultValue="0%">

<ref name="percent"/>

</attribute>

</optional>

</define>

#### Extrusion Number Of Line Segments

The draw:extrusion-number-of-line-segments attribute specifies the number of line segments that should be used to display curved surfaces. The higher the number the more line segments are used.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="draw:extrusion-number-of-line-segments"

a:defaultValue="30">

<ref name="integer"/>

</attribute>

</optional>

</define>

#### Extrusion Light Face

The draw:extrusion-light-face attribute specifies if the front face of the extrusion responds to lightning changes.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="draw:extrusion-light-face" a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Extrusion First Light Harsh

The draw:extrusion-first-light-harsh attribute specifies if the primary light is harsh.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="draw:extrusion-first-light-harsh"

a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Extrusion Second Light Harsh

The draw:extrusion-second-light-harsh attribute specifies if the secondary light is harsh.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="draw:extrusion-second-light-harsh"

a:defaultValue="true">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Extrusion First Light Level

The draw:extrusion-first-light-level attribute specifies the intensity for the first light.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="draw:extrusion-first-light-level"

a:defaultValue="66%">

<ref name="percent"/>

</attribute>

</optional>

</define>

#### Extrusion Second Light Level

The draw:extrusion-second-light-level attribute specifies the intensity for the second light.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="draw:extrusion-second-light-level"

a:defaultValue="66%">

<ref name="percent"/>

</attribute>

</optional>

</define>

#### Extrusion First Light Direction

The draw:extrusion-first-light-direction attribute specifies the direction of the first light.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="draw:extrusion-first-light-direction"

a:defaultValue="(5 0 1)">

<ref name="vector3D"/>

</attribute>

</optional>

</define>

#### Extrusion Second Light Direction

The draw:extrusion-second-light-direction attribute specifies the direction of the second light.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="draw:extrusion-second-light-direction"

a:defaultValue="(-5 0 1)">

<ref name="vector3D"/>

</attribute>

</optional>

</define>

#### Extrusion Metal

The draw:extrusion-metal attribute specifies if the surface of the extrusion object looks like metal.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="draw:extrusion-metal" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

#### Extrusion Shade Mode

The dr3d:shade-mode attribute defines how the lighting is calculated for rendered surfaces

* flat: lighting is calculated by one surface normal.
* phong: lighting is calculated by interpolating the surface normals over the surface.
* gouraud: lighting is calculated by interpolating the color calculated with the surface normals at each edge.
* draft: surfaces are not lit and drawn as wireframe only.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="dr3d:shade-mode" a:defaultValue="flat">

<choice>

<value>flat</value>

<value>phong</value>

<value>gouraud</value>

<value>draft</value>

</choice>

</attribute>

</optional>

</define>

#### Extrusion Rotation Angle

The first value of the draw:extrusion-rotation-angle specifies the rotation about the x-axis. The second value of the draw:extrusion-rotation-angle specifies the rotation about the y-axis. The rotation about the z-axis is specified by the rotate angle of the draw:transform attribute.

The order of the rotation is: z-axis, y-axis and then x-axis.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="draw:extrusion-rotation-angle" a:defaultValue="0 0">

<list>

<ref name="double"/>

<ref name="double"/>

</list>

</attribute>

</optional>

</define>

#### Extrusion Rotation Center

The draw:extrusion-rotation-center attribute specifies the position of the rotation center in terms of shape size fractions, if it is omitted then the geometrical center of the shape is used.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="draw:extrusion-rotation-center">

<ref name="vector3D"/>

</attribute>

</optional>

</define>

#### Extrusion Shininess

The draw:extrusion-shininess attribute specifies the shininess of a mirror.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="draw:extrusion-shininess" a:defaultValue="50%">

<ref name="percent"/>

</attribute>

</optional>

</define>

#### Extrusion Skew

The draw:extrusion-skew attribute specifies the skew amount and skew angle of an extrusion. Skew settings are only applied if the attribute dr3d:projection has the value parallel.

The first parameter represents the skew amount in percent, the second parameter specifies the skew angle.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="draw:extrusion-skew" a:defaultValue="50 45">

<list>

<ref name="double"/>

<ref name="double"/>

</list>

</attribute>

</optional>

</define>

#### Extrusion Specularity

The draw:extrusion-specularity attribute specifies the specularity of an extrusion object.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="draw:extrusion-specularity" a:defaultValue="0%">

<ref name="percent"/>

</attribute>

</optional>

</define>

#### Extrusion Projection Mode

The dr3d:projection attribute specifies if the projection mode is perspective or parallel.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="dr3d:projection" a:defaultValue="parallel">

<choice>

<value>parallel</value>

<value>perspective</value>

</choice>

</attribute>

</optional>

</define>

#### Extrusion Viewpoint

The draw:extrusion-viewpoint attribute specifies the viewpoint of the observer as an 3D point. The attribute's value syntax is similar to vector3D, solely a unit is following each parameter. An example for a 3D point is: “(1cm 1cm 0m)”.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="draw:extrusion-viewpoint"

a:defaultValue="3.5cm -3.5cm 25cm">

<ref name="point3D"/>

</attribute>

</optional>

</define>

<define name="point3D">

<data type="string"/>

</define>

#### Extrusion Origin

The draw:extrusion-origin attributes specifies the origin within the bounding box of the shape in terms of the shape size fractions.

The first parameter represents the horizontal origin, a value of -0.5 represents the left side of the shape, a value of 0 represents the center of the shape, a value of 0.5 represents the right side of the shape.

The second parameter represents the vertical origin, a value of -0.5 represents the top side of the shape, a value of 0 represents the center of the shape, a value of 0.5 represents the bottom side of the shape.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="draw:extrusion-origin" a:defaultValue="0.5 -0.5">

<list>

<ref name="double"/>

<ref name="double"/>

</list>

</attribute>

</optional>

</define>

#### Extrusion Color

The draw:extrusion-color attribute specifies if an extrusion color is used. The extrusion color is then defined by the draw:secondary-fill-color attribute specified in the custom shape's graphic style.

<define name="draw-enhanced-geometry-attlist" combine="interleave">

<optional>

<attribute name="draw:extrusion-color" a:defaultValue="false">

<ref name="boolean"/>

</attribute>

</optional>

</define>

### Enhanced Geometry - Path Attributes

#### Enhanced Path

The draw:enhanced-path attribute specifies a path similar to the svg:d attribute of the <svg:path> element. Instructions such as moveto, lineto, arcto and other instructions together with its parameter are describing the geometry of a shape which can be filled and or stroked. Relative commands are not supported.

The syntax of draw:enhanced-path attribute is as follows:

* Instructions are expressed as one character (e.g., a moveto is expressed as an M).
* A prefix notation is being used, that means that each command is followed by its parameter.
* Superfluous white space and separators such as commas can be eliminated. (e.g., “M 10 10 L 20 20 L 30 20” can also be written: “M10 10L20 20L30 20”
* If the command is repeated multiple times, only the first command is required. (e.g., “M 10 10 L 20 20 L 30 20” can also be expressed as followed “M 10 10 L 20 20 30 20”
* Floats can be used, therefore the only allowable decimal point is a dot (“.”)

The above mentioned rules are the same as specified for the <svg:path> element.

A parameter can also have one of the following enhancements:

* A “?” is used to mark the beginning of a formula name. The result of the element's draw:formula attribute is used as parameter value in this case.
* If “$” is preceding a integer value, the value is indexing a draw:modifiers attribute. The corresponding modifier value is used as parameter value then.

Following notation is used in the table below:

* (): grouping of parameters
* +: 1 or more of the given parameter(s) is required

**Example** for a custom-shape that uses the draw:enhanced-path to describe a pie-chart whose top right quarter segment is taken out:

<draw:custom-shape

svg:width="10cm" svg:height="10cm" svg:x="0cm" svg:y="0cm">

<draw:enhanced-geometry svg:viewBox="0 0 10 10"

draw:enhanced-path="V 0 0 10 10 10 5 5 0 L 5 5 Z N">

</draw:enhanced-geometry>

</draw:custom-shape>

The following commands are supported:

| Command | Name | Parameters | Description |
| --- | --- | --- | --- |
| M | moveto | (x y) + | Start a new sub-path at the given (x,y) coordinate. If a moveto is followed by multiple pairs of coordinates, they are treated as lineto. |